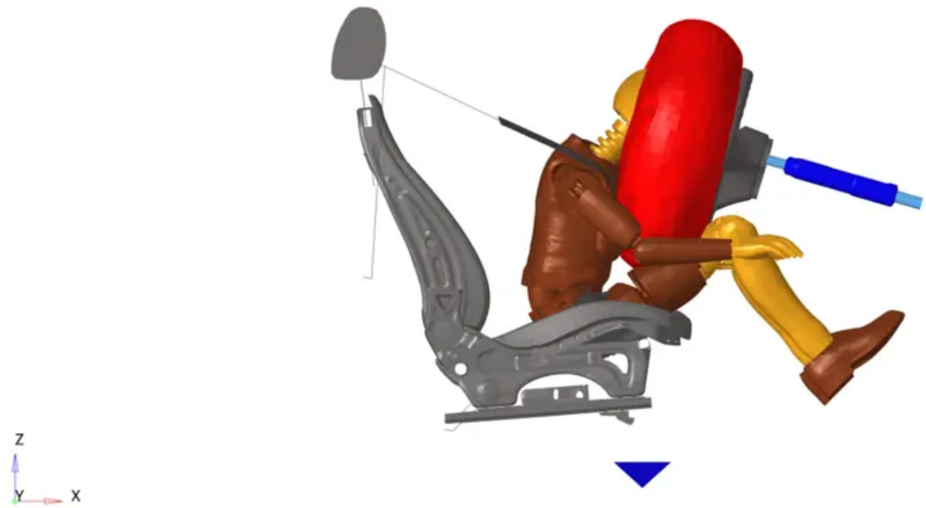
CHALMERS
UNIVERSITY OF TECHNOLOGY



An Optimization and Cluster based Approach to Lookup Tables in Design of Adaptive Restraint Systems

PRANAV ANAND

RAHUL DASIKATTE ASHOK KUMAR

DEPARTMENT OF INDUSTRIAL AND MATERIAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

An Optimization and Cluster based Approach to Lookup Tables in Design of Adaptive Restraint Systems

PRANAV ANAND
RAHUL DASIKATTE ASHOK KUMAR

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Material Sciences
Chalmers University of Technology
Gothenburg, Sweden 2024

An Optimization and Cluster based Approach to Lookup Tables in Design of Adaptive Restraint Systems
PRANAV ANAND
RAHUL DASIKATTE ASHOK KUMAR

Industrial Supervisors: Lara Wehrmeyer and Henrik Balague, Autoliv Sverige AB
University Supervisor: Mohammad Arjomandi Rad
Examiner: Ola Isaksson, Product Development

Master's Thesis 2024
Department of Industrial and Material Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Abstract

The timely deployment of vehicle restraint systems is crucial in mitigating the impact of collisions and protecting occupants in the affected vehicles. The level of protection can be further enhanced with the use of adaptive restraint systems, by adjusting the force and timing of seatbelts and airbags based on factors such as vehicle speed, occupant size, and seating position. Virtual testing methods can help identify areas of improvement of adaptive systems by evaluating its performance across a range of crash scenarios. While finite element simulation models and data-based surrogate models have been used in literature for restraint system development, data structures such as lookup tables containing restraint settings offer potential to accelerate the design and deployment of adaptive restraint systems. To implement this, firstly, metamodels using Gaussian process regression were developed to predict specific occupant kinematics and injury risks in frontal collisions between vehicles with varying crash configurations. Furthermore, the optimal restraint settings for each crash configuration was identified using a genetic algorithm, taking into account the injury risk predictions from the metamodels. The frontal collisions were categorized based on crash pulse intensities and then represented in a static lookup table for quick retrieval of restraint settings based on a crash configuration. The restraint settings obtained from the optimization were validated with real-world equivalent settings, exhibiting lower injury risks in low and medium-speed crashes for the concerned vehicle model. Overall, the research presented demonstrates the application of look-up tables as a tool for development and operation of adaptive restraint systems. Furthermore, the combination of look-up tables with machine learning techniques could be scaled to suit the complexity of the engineering problem.

Keywords: Metamodel, Genetic algorithm, Restraint settings, Crash scenario, Injury risk, Lookup table, Gaussian process, Clustering, Machine learning.

Acknowledgements

We would like to express our sincere gratitude towards our supervisors, Lara Wehrmeyer and Henrik Balague from the Simulation Engineering department at Autoliv Sverige AB, Vårgårda, for their unwavering guidance and support throughout our thesis project. Their valuable feedback and technical insights have not only helped us in learning new technical skills, but also helped ensure timely progress of the thesis. We would also like to extend our gratitude towards our project manager, Mika Himiläinen, for making the necessary arrangements and encouraging us throughout the course of the research.

We would also like to thank the Industrial and Material sciences department at Chalmers for their unconditional support and assistance throughout this thesis. We extend our sincere gratitude towards our supervisor, Mohammad Arjomandi Rad, for his resourceful feedback and encouraging us to explore deeper into our domain. His contribution was crucial in guiding us towards fulfilling our project goals and research. Finally, we would like to extend our appreciation towards our examiner Ola Isaksson, for supervising us and assisting towards the completion of our thesis.

Pranav Anand & Rahul Dasikatte Ashok Kumar , Gothenburg, 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADAS - Advanced Driver Assistance Systems
AI - Artificial Intelligence
BO - Bayesian Optimization
BMI - Body Mass Index
CAE - Computer Aided Engineering
CGA - Chaos Genetic Algorithm
CRR - Conceptual Rainfall Runoff
DAB - Driver Airbag
DAI - Diffuse Axonal Injury
DAMAGE - Diffuse Axonal Multi-Axial General Evaluation
DOE - Design of Experiments
ECU - Electronic Control Unit
FEM - Finite Element Model
GA - Genetic Algorithm
GPR - Gaussian Process Regressor
HBM - Human Body Model
HIC - Head Injury Criterion
KNN - K Nearest Neighbours
LASSO - Least Absolute Shrinkage and Selection Operator
LHS - Latin Hypercube Sampling
LLA - Load Limiter Adaptive
LWPR - Locally Weighted Projection Regression
MAE - Mean Absolute Error
MCS - Monte Carlo Sampling
MSE - Mean Squared Error
NCAP - New Car Assessment Program
OLS - Ordinary Least Squares
RBF - Radial Basis Function
RLS - Recursive Least Squares
RMSE - Root Mean Square Error
RSM - Response Surface Method
SA - Simulated Annealing
SOTA - State of the Art

SRSM - Successive Response Surface Method
SVM - Support Vector Machines
SW - Steering Wheel
TBI - Traumatic Brain Injury
TTF - Time to Fire

Contents

List of Figures

List of Tables

1
Introduction

The numerous risks associated with vehicular travel necessitate the need and focus on occupant safety and crash protection in the event of collisions. Road accidents are one of the highest causes of fatalities in the world after health-related complications, with over 1 million lives lost annually due to collisions (Roser, 2023). Therefore, robust safety systems in vehicles play an important role in reducing the risk of injuries and fatalities to the driver and other occupants. Recent advancements in vehicle safety technology, along with periodic revisions to occupant protection standards have contributed to the development of safer systems. Modern vehicles are equipped with both active and passive safety systems such as advanced driver assistance systems (ADAS), seatbelts, and airbags, aiding in crash avoidance and occupant protection in the pre-crash and in-crash phases.

Integrated safety systems are a combination of the capabilities of threat assessment and intervention systems along with occupant protection systems like seatbelts and airbags. Restraint systems have predefined settings and thresholds and act based on the vehicle speed, braking, and steering inputs, and other parameters. Depending on the situation and set of triggers, the seatbelt and airbag will actuate with a certain trigger time and force setting , dissipating the crash energy and reducing the loads on the occupant. Current restraint systems are semi-adaptive, meaning that restraint settings can be adjusted based on specific scenarios but not all cases. However, the restraint systems can be improved by adapting to a crash scenario with the help of other sensors in the vehicle such as a Radar and Lidar which help scan the external environment around the vehicle (Eppinger et al., 2017). Based on the vehicle surroundings, the restraint settings can be adjusted in real-time in anticipation of a crash event. The restraint settings also differ from crash to crash due to some factors such as vehicle speed, occupant size, and occupant seating position. The work carried out in the thesis focuses on formulating a lookup table with the help of machine learning tools in the development of adaptive restraint systems. Lookup tables have been utilized in various computer applications such as data and image processing to perform calculation and conversion related tasks (Guardiola et al., 2013). The table stores pre-calculated values of a desired output in connection with the input in the working memory of the system. The desired output values can be accessed through a direct lookup using the memory, saving time and avoiding extra computation. In the context of adaptive restraint systems, the lookup table can be utilized in determining the optimal restraint settings quickly to protect the occupants involved in a collision scenario. Additionally, the pre-calculated restraint settings in the lookup table provides a foundation for design and development of future adaptive safety systems offering scalability based on system complexity.

1.1 Objective

Compared to traditional restraint systems, adaptive restraint systems can dynamically adjust the restraint parameters in response to the crash situation to improve occupant protection. By using external vehicle sensors to scan the vehicle surroundings, an adaptive system can tailor specific settings of the seatbelts and airbags to reduce the risk of injury and strain to the occupants. However, the system's effectiveness is dependent on a robust framework that contains a set of restraint settings for different vehicle speeds and other parameters to improve its adaptability for various collision scenarios. One of the main goals of this thesis is to develop a framework for an adaptive restraint system with the help of a lookup table. The table will present a set of restraint settings for multiple vehicle velocities and accelerations, which have been obtained using machine learning tools. The thesis work has been carried out with a series of objectives that has been outlined below:

1. Develop and train metamodels to predict specific kinematic and injury risks from a dataset containing simulated crash data for two identical vehicle models involved in frontal collisions.
2. Optimize the restraint settings for different collision scenarios using the injury risk predictions from the trained metamodels.
3. Represent the restraint settings corresponding to vehicle velocities in a lookup table and group them according to crash pulses.

With the steps stated above, the cluster-based lookup table presents a potential as a tool for development and operation of adaptive restraint systems. There were several challenges that arose during the thesis in pursuit of our objectives and each of them have been addressed by formulating some research questions that guided our work. These include:

- Can model-based optimization of restraint system settings help in reducing the injury outcomes of occupants during a crash?
- How can the complexity of restraint systems be reduced by grouping and visualizing optimized restraint settings for different crash scenarios?
- How can lookup tables be combined with other machine learning based tools in selecting the optimized restraint settings for crash scenarios?

1.2 Background Study

The importance of vehicle safety in crash and occupant protection has been established in the previous sections. Modern vehicles utilize pre-crash elements (active safety) and in-crash elements (passive safety) in order to avoid crashes and provide adequate protection to the occupants. The assessment of these systems is tested and verified by an independent body like the Euro NCAP on the five-star safety rating system, providing estimates of the level of occupant protection offered by the vehicle (Van Ratingen et al., 2016). Vehicles with three or four adult occupant stars are approximately 30% safer, compared to two-star cars or cars without a Euro NCAP score. Moreover, the relative predicted risk of injuries or fatalities were reduced by 12% for every star in the Euro NCAP star rating (Van Ratingen et al., 2016). Along with physical testing, an alternative approach using virtual testing methods have been explored by various car manufacturers to analyse and improve vehicle safety.

1.2.1 Metamodel studies

Currently, manufacturers utilise CAE simulation tools to analyse crash scenarios on virtual vehicle models since they are cost-effective in comparison to physical testing methods and allow for flexibility in running different crash cases. However, running a single crash case could take up to 20 hrs to simulate and the computational time and resources required increases exponentially as more cases are being analysed (Hay et al., 2023). Hence data based surrogate models were studied to capture the underlying trends from the simulations and to make predictions based on the data (Hay et al., 2023). Different types of data based surrogate models, including Ordinary Least Squares (OLS), Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Machines (SVM), Neural Networks, Random Forest, and Ensemble methods were used to approximate human body model responses in restraint design considerations (Joodaki, Gepner, and Kerrigan, 2021). These models were assessed on data involving a human body model (HBM) modified to two types of anthropometries, obese (BMI = 35 kg/m2) and normal (BMI = 25 kg/m2) (Joodaki, Gepner, Lee, et al., 2021). They were placed on a sled buck with the standard vehicle restraint systems with appropriate modifications for a frontal rigid barrier test at 56 km/h. The ensemble model, which was a combined average of all the machine learning models used, performed the best in estimating the human body responses (Joodaki, Gepner, and Kerrigan, 2021). The regression models presented in these studies have their own underlying mathematical formulations and work well under the assumption that the data also follows similar patterns. With the complexities involved in data types and ranges in the Design of Experiments (DoE) dataset, a flexible function that maps the inputs to the outputs was desired. A gaussian process regressor metamodel was used to capture the variations in non-linear data and was able to accurately model the relationship between restraint setting parameters and human body responses (Hay et al., 2023). The model estimates the mean and standard deviation of each data point, and a covariance matrix determines the distance between each point and fits a line or curve according to the correlation between each point (J. Wang, 2023). Considering the variability of the predictions from metamodels, three different methods – D-filling, latin hypercube sampling (space-filling design), and adaptive sampling techniques were used to sample the dataset. The samples generated using the space-filling method accounted for non-linearities and constraints within the dataset while ensuring that the overall distribution was evenly spaced (Hay et al., 2023). A GPR based spatiotemporal model was developed to study electrical wave propagation in the human heart, replacing time-intensive simulation models for research cardiac function (Hu et al., 2020). GPR was also used in a multi-fidelity data fusion model and compared with a Cokriging model. The results showed that the gradient-enhanced Gaussian process metamodel showcased better performance in predicting quantities of interest and their respective gradients (Deng et al., 2020). The performance of non-parametric regression models like GPR and locally weighted projection regression (LWPR) were compared in terms of approximation of a torque control function for a robotic arm. The SARCOS robot arm data was used for model approximation and GPR was preferred in real-time control due to its accuracy (Nguyen-Tuong et al., 2008). The studies presented have attributed the accuracy, versatility, and applicability of the regression model in several engineering disciplines for capturing complex relationships within the data.

1.2.2 Optimization studies

In the context of optimizing restraint settings, combining the predictive capabilities of the metamodel with machine learning-based optimization algorithm can lead to improved results. A heuristic approach like the genetic algorithm was utilized to enhance an adaptive vehicle restraint system. The algorithm demonstrated 89% and 42% time savings in attaining optimal restraint settings in the driver and passenger's side for a vehicle compared to multi-body simulations, however the validity of the optimized settings was not discussed (Yeh et al., 2005). Another study used different types of optimization strategies such as Gradient descent, Random search, Genetic algorithm, and Response surfaces were utilized to solve non-linear problems. Based on the approximation in finding the optima, the genetic algorithm and a response surface method (RSM) using neural networks were used to optimize an adaptive restraint system for frontal collisions with four types of load cases concerning male and female dummies in belted and unbelted conditions. The Successive response surface method (SRSM) provided better results compared to the genetic algorithm, meeting the constraints for all load cases (van den Hove et al., 2005). In other fields, two global optimization (Genetic algorithm and Simulated Annealing) and deterministic methods were compared in an inverse function to approximate the parameters of a dynamic model for oceanic cycling rates. The gradient requirement for deterministic methods limited the search for the solution while the simulated annealing algorithm was stuck in local minima. However, the genetic algorithm quickly estimated the eight parameters and converged to the optimal solution with lower computational requirements in comparison to other methods (Athias et al., 2000). A hybrid genetic algorithm combining simulated annealing was employed to estimate the conceptual rainfall runoff (CRR) in the Shuangpai reservoir using flood data from 2000-2006 and compared with a chaos genetic algorithm (CGA) and a standard genetic algorithm. The hybrid algorithm combining SA performed better than the other two methods in solving the CRR problem (W.-C. Wang et al., 2012). The presented studies demonstrate the flexibility, speed, and lesser computational requirements of the genetic algorithm in finding an optimal solution to the objective function. The results from optimization have the potential to enhance the adaptability and performance of restraint systems based on the input variables of the vehicle.

1.2.3 Lookup table studies

Optimal restraint settings for different crash scenarios can be obtained with the help of an optimization approach. The optimized settings can be stored in data structures and similar methods and used in decision-making and other engineering applications. Expert systems and knowledge-based systems are types of Artificial Intelligence (AI) systems that emulate human decision-making abilities based on domain knowledge and reasoning abilities to solve complex problems. For example, an expert system approach was utilized to build a prototype system for ergonomic workplace design and evaluation, using modular knowledge bases to include a variety of ergonomic factors (Jung, 1988). Expert systems were also tested on its performance in dynamic control task, highlighting the importance of a good model for process dynamics for expert operators (Hayes-Roth, 1984). On the other hand, embedded control systems such as automotive systems, robotics, and electronic systems use lookup tables to emphasize speed and control on repetitive tasks by storing pre-computed values of inputs and outputs (Bengtsson, 2012). Although expert systems, knowledge systems, and other AI-based systems can improve through continuous learning, lookup tables are preferred due to its simplicity, speed, and efficiency through memory recall. In automotive applications, a dynamic lookup table was used to map engine exhaust variables in a Diesel engine such as λ^{-1} and NO_x over a driving profile. The table was updated using offline Kalman filter gains in various scenarios, such as a sporty driving situation. Following an initial sensor data error, the table accurately matched engine variables over extended running periods (Guardiola et al., 2013). Alternatively, recursive least squares (RLS) based adaptive lookup tables were used to map the uncertain volumetric efficiency of a gasoline engine for a simulated driving cycle. The RLS-based adaptive lookup table had better adaptability in mapping the volumetric efficiency model and was validated by lower root mean square (RMS) error values compared to a static table (Lochrie et al., 2021).

1.2.4 Summary

From the background study, the researchers have demonstrated the benefits of utilising data-based surrogate models in capturing the underlying trends from CAE simulations with significant computational time savings and relatively low loss of accuracy. In studies relating to vehicle safety applications, these metamodels were used to predict injury responses and compared against simulations, estimating the error margins and to also develop a suitable design framework for restraint setting optimization with the help of applicable optimization algorithms. Lookup tables have also been used in engineering applications such as mapping engine inputs to different pollutants in a range of driving scenarios and in estimating the volumetric efficiency of an engine. The thesis work carried out aims to develop a cluster-based lookup table for selecting the most optimal restraint settings for a diverse range of frontal collision scenarios, aiding in future design and operation of adaptive restraint systems.

1.3 Limitations

- The thesis is limited to frontal collisions on account of model simplification. There are variations in vehicle overlap, although the impact angles remain the same. With rear-end and far-side collisions, the model needs to consider additional complexities such as impact angles, injury risks, and protection systems such as side curtain airbags.
- Data-driven models are generally trained on simulation data and the output from these models are approximations of the original dataset accompanied by a certain error. Therefore, the error margin of the metamodels and optimization should be considered while interpreting injury outcomes.
- Another limitation of this approach would be that the proposed optimization and lookup table approach is limited to one specific vehicle model. Vehicle models differ in crash structure, shape, safety equipment, and crash kinematics. The outcomes from one model may not be applicable in another model, which necessitates model re-training and tuning to reflect the new vehicle model.
- The influence of human variabilities were not considered in the thesis. Factors such as occupant size, age, gender interact differently with vehicle restraint systems and require further tuning and model considerations while optimizing restraint settings.

2 Theory

2.1 Restraint Systems

Restraint systems in vehicles mainly comprise airbags and seatbelts, often combined with pre-tensioners and load limiters to adjust seatbelt tension and protect the occupants. They play a critical role in restraining passenger movements and safeguarding them from life-threatening injuries by distributing forces during collisions (Yeh et al., 2005). Adaptive restraint systems are vehicle safety systems that can dynamically adjust and modulate the force and timing of seatbelts and airbags based on factors like vehicle speed, occupant position, and size to provide enhanced protection (Eppinger et al., 2017). In the automotive industry, restraint settings refer to the configuration or calibration of safety systems present in the automobile. Crash scenarios, which refer to simulated or real-life situations involving vehicle collisions are designed to assess and evaluate the vehicle's safety system and structural integrity under various conditions. Proper tuning of the restraint settings will ensure optimal performance during different crash scenarios while maximizing occupant protection during vehicular accidents (Lemmen et al., 2012).

2.2 Machine Learning Models

The selection of an appropriate metamodel for representing the data is a crucial aspect for making accurate predictions that also represent the trends of the simulations. The most suitable models for predicting data points based on given input data are regression models. Regression involves extrapolating data points in a dataset using an underlying mathematical function (Sarker, 2021). Various regression models like linear, polynomial, and logistic regression aid in point identification based on the mathematical functions specific to each regression type. Depending on the type of regression approach, the model assumes that the input and output variables follow a certain mathematical function, which is used for predicting the next set of data points. However, the design of experiments (DOE) used in the thesis contains different variable types and ranges, which made it challenging for selecting fixed functional models to accurately capture the relationship between the chosen design variables and targets.

According to Joodaki, Gepner, and Kerrigan, 2021, surrogate models such as Ordinary Least Squares (OLS), Least Absolute Shrinkage and Selection Operator

(LASSO), Support Vector Machine (SVM), Neural Networks, and Random Forests were used to approximate injury responses in automotive safety applications. However each model has different constraints and is suitable for solving more comprehensive problems.

2.2.1 Gaussian Process Regression (GPR)

Gaussian Process Regression does not follow a fixed relationship with data points unlike other regression models described above. Depending on the positions of data points scattered on a plot, the GPR models the relationship between the inputs and outputs by evaluating the mean and variance of each point, thereby fitting multiple curves to the data points based on the euclidean distance and variance between each point described by the covariance functions as shown in figure 2.1 (Williams and Rasmussen, 2006). The model also considers prior knowledge of the dataset to provide more accurate predictions than other regression models and its associated uncertainties, making it suitable for handling non-linear relationships and other constraints within the data points (Hay et al., 2023).

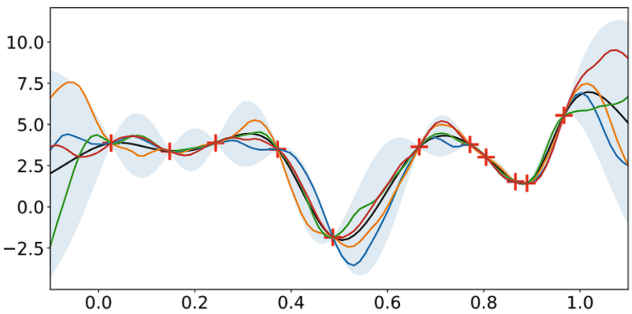


Figure 2.1: Curve fitting using GPR (Image Source: J. Wang, 2023)

The model also accounts for the amount of inherent noise present within the data. While other methods may assume a particular noise distribution or include added noise, the GPR is parametrized by noise level which controls the amount of added noise in the data along with the uncertainties present (Scikit-Learn, 2024a). This aspect is beneficial when working with data affected by measurement errors and sensor noise.

Although being a flexible regression model, it faces difficulties with larger datasets. It works well with hundreds of data points but struggles with much larger datasets containing thousands or millions of data points and is limited by computational resources (J. Wang, 2023). Furthermore, it contains different hyperparameters that require tuning, playing an influence on the model performance and predictive capabilities (Williams and Rasmussen, 2006).

2.2.1.1 Kernels

Kernels are mathematical functions that helps in determining the similarity or correlation between any given data points. They influence the model's predictive abilities by controlling the functional parameters such as shape, smoothness, and function behaviour (Scikit-Learn, 2024a). For our metamodels, we used three different kernel types, namely, radial basis functions (RBF), constant kernel, and white noise kernel. These specific kernels were chosen to capture the trends and variations in the dataset. The RBF kernel is used to analyze the nonlinear relationships in the data, while the constant kernel is used to find the global mean. Modeling the inherent variations of data in the dataset is made easier by the white noise kernel. In addition to improving the accuracy, we make the model more robust with the use of these kernels. A more detailed overview of the three kernels are shown below:

Radial basis function kernel (RBF): The Radial basis function kernel, also known as the squared exponential kernel is one of the most widely used kernels in the gaussian process regression model. It is represented by the equation shown below:

$$k(xi, xj) = exp\left(-\frac{d(xi, xj)^2}{2l^2}\right) \tag{2.1}$$

The radial basis function kernel (RBF) is mainly controlled by the length scale parameter which controls the smoothness of the function (Scikit-Learn, 2024a).

Constant kernel:
The constant kernel works similarly to a constant term in an equation, implying that it adds a standard offset value to another kernel.It helps in scaling the magnitude of other kernels to desired values (Scikit-Learn, 2024a). It is defined as:

$$k(\ xi,\ xj) = constant\_value \quad \forall x1, x2 \tag{2.2}$$

White Noise kernel:
The white noise kernel, unlike the constant kernel, primarily focuses on the noise inherent in data and aids in estimating the white noise present in a model. For similar data points, this kernel can provide specific noise measurements for each point (Scikit-Learn, 2024a).

By tuning the model's hyperparameters, the white noise kernel can account for the noise present in the data, thus enabling more effective handling of noisy datasets. It is characterized by the equation below:

$$\mathbf{k}(\mathbf{x_i}, \mathbf{x_j}) = \begin{cases} \text{noise}_{\text{level}} & \text{if } x_i = x_j \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Section 3.3 describes how the GPR metamodel is used for training with the help of the kernel functions. It also discusses about the selection and tuning of the model hyperparameters to improve its accuracy. The accuracy of the metamodel is evaluated with the help of error metrics.

2.2.1.2 Error Metrics

Error metrics are statistical tools which help in evaluating the predictive capabilities of a metamodel. The performance of the model is characterized by the difference of the predicted values to the actual values in the dataset. The smaller the difference between the actual and predicted values, the lower the error (Plevris et al., 2022). The error metrics used to measure the performance of the GPR are depicted below:

i) Mean Absolute Error (MAE)

The mean absolute error is defined as the average absolute difference error values between the predicted and the actual values (Plevris et al., 2022). It is expressed in the equation shown below:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.4}$$

ii) Mean Squared Error (MSE):

The mean squared error determines the average difference of the squares between the predicted and actual values. Due to the presence of the square term in the formula, larger errors are penalized much more, possibly indicating the presence of outliers in the data (Plevris et al., 2022). The equation for MSE is shown below:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.5}$$

iii) Root Mean Squared Error(RMSE):

The root mean squared error is the square root of the mean squared error and it gives an estimation of the standard deviation between the predicted and actual values (Plevris et al., 2022). Larger errors are penalized in an identical fashion to MSE, however, the error can be compared more easily since the error term contains the same units as the data point being analyzed. The formula for calculating RMSE is demonstrated below:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.6}$$

iv) **Coefficient of Determination(R2 Score)**

The coefficient of determination measures the proportion of variance of the model's dependant variable in relation to the independent variable. It's value lies between 0 and 1 and the closer the value is to 1, the better performing a model is (Plevris et al., 2022). The coefficient of determination is calculated using this formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.7}$$

Section 4.1 demonstrates how metamodels are evaluated for their accuracy, while Table 4.1 showcases the results obtained after using the error metrics.

2.3 Optimization

In the context of machine learning methods, optimization is generally used to find the best settings (optimal) for a model to improve its prediction capabilities. The metamodel is trained to identify patterns from data and make predictions, although its parameters need to be tuned to improve predictions. An optimization algorithm helps in determining the most suitable model parameters to minimize errors while making predictions. This is a type of optimization used in finding suitable model hyperparameters, however, it can also be utilized to find the best set of decision variables or input parameters for maximizing or minimizing the target variables predicted by the metamodels. The optimization algorithm is often combined with metamodels to evaluate the solution to complex objective functions, which might require more computational resources, allowing for faster searches in the solution space (Yeh et al., 2005). By balancing the optimization in terms of exploration and exploitation, we can have a better understanding of the relationship between the data to obtain optimal results. This can aid in understanding the system behaviour across various scenarios, thereby increasing the models reliability and decision making to achieve the required objectives.

The goal of optimization is to select the most suitable decision variables to minimize the target variables. In our application, the objective is to help select the most optimal restraint settings to reduce the injury risks to the occupants. Multiple optimization methods and types have been explored to carry out the task, although the dataset consisted of multiple variables with discrete and continuous ranges. Moreover, since the minimization process needs to be applied for multiple target variables, it necessitates a more flexible optimization method.

To achieve this, different stochastic optimization methods that supported a Gaussian Process Regressor (GPR) metamodel have been tested, including Simulated Annealing (SA), and Bayesian Optimization (BO). However, both methods were unsuccessful in converging to the optimal solution since it did not account for multiple variable types. The genetic algorithm was the most suitable model since it supports multiple variable types as inputs and also explores a broader solution space. A more detailed explanation of the algorithm is illustrated in section 2.3.1.

2.3.1 Genetic Algorithm

Genetic algorithm is an optimization process that follows the principles of natural selection and evolution analogous to the survival of the fittest concept in biology (Gad, 2023). These algorithms are used to solve complex engineering problems where there is a requirement to find the most optimal value.
The working of genetic algorithm can be shown in Fig 2.2:
Out of the many libraries available to use for GA in python, we make use of PyGAD which is an easy to use library specifically implemented for genetic algorithm in python.

Initialization: The algorithm begins by generating an initial population of values from the inputs which might consist of a possible set of solutions.

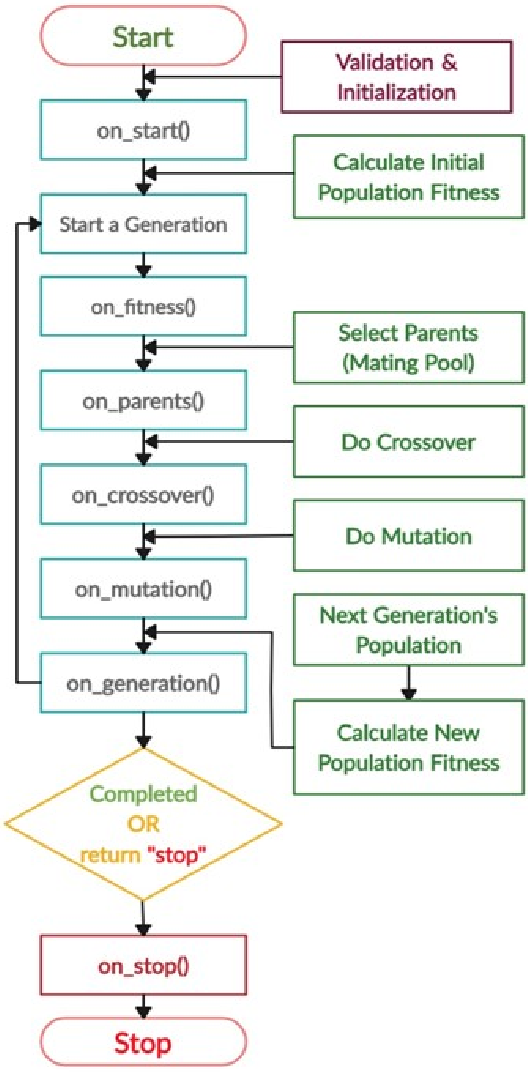


Figure 2.2: Genetic Algorithm Process Flowchart (Gad, 2023)

Fitness: Individuals from the population are evaluated on their fitness based on conditions defined in the user-defined fitness function. It primarily consists of the constraints used to obtain the desired output.

Selection: The selection operator selects the best individuals to become parents for the next generation on the basis of its fitness score. This is carried out to ensure propagation of good genes and promote diversity of solutions.

Crossover: The crossover operation selects individuals from the current population, often referred to as parents, and combines their genetic information to create offspring for the next generation.

Mutation: A random selection of individuals are subjected to mutations or changes in their genetic code to enable diversity and greater exploration of the solution space.

Replacement: The replacement operator selects individuals for the next generation by retaining the offspring with the highest fitness scores, and potentially some of the parents from the previous generation if their fitness scores are also high.

Termination: The process is repeated for the specified number of generations until the values converge and reach saturation where we obtain our optimal value.

Since Genetic algorithm is a stochastic and population based optimization method, it is one of the best methods that could help in finding the global optimum.

When dealing with noisy, discontinuous, or ill-defined objective functions, genetic algorithms are resilient and flexible (Yeh et al., 2005). They can handle uncertainty and variance in the optimization landscape thanks to their diversity preservation methods and population-based search approach.

Genetic algorithms (GA) are very versatile because they are not subject to domain knowledge and assumptions, so they are useful in engineering, finance, biology, and other areas. Additionally, the genetic algorithms' selection, crossover, and mutation operators allow iterative improvements and rapid navigation across promising regions by balancing exploration and exploitation within the solution space. A wide range of fields can be solved using genetic algorithms, which are powerful optimization methods. They are indispensable tools for effectively and efficiently managing real-world optimization tasks due to their ability to detect global optima as well as their scalability, adaptability, and tolerance to noise.

In order to use GA we needed an initial population as our input to help us generate suitable solutions. Stratified sampling method like that of Latin hypercube sampling method was selected to help generate unique crash scenario samples to be used here. It is further discussed in more detail in section 2.3.2 and its methodology is explained in section 3.4.1. The methodology for carrying out the optimization process is explained in section 3.4 with the help of the process flow in figure 3.6.

2.3.2 Stratified Sampling

Evolutionary algorithms such as GA require an initial population to carry out the optimization process. The input variables used in the dataset have discrete and continuous values within certain constraints that need to be sampled. To account for these differences, a stratified sampling algorithm has been used to create distinct input samples for optimization. Stratified sampling typically involves dividing the entire population or distribution into distinct subgroups or strata that are non-overlapping and collectively exhaustive (Shields and Zhang, 2016). Monte Carlo Sampling (MCS) and Latin Hypercube Sampling (LHS) are two types of sampling methods that generate unique values for a normal distribution. MCS is comparable to simple random sampling where 'n' random values are populated based on an already known probability distribution (Genie, 2024). Due to the randomized process, there are chances of values being repeated during sampling. In the case of latin hypercube sampling, the basis for creation of a sample is explained by the concept of a latin square, where each input sample is divided into 'n' equal intervals for a set of ‘k’ variables. The overall distribution will contain a set of k x n values. Each individual square in the k x n distribution is considered to be a latin square since it contains a unique value of the sample per row and column in the distribution.

The algorithm samples random values that from each square, ensuring that the values do not coincide with other dimensions each time it is sampled from the dataset. While the square is represented in a two-dimensional form, it can be extended to any number of dimensions as required, where each sample in a dimension contains only one value in the axis aligned hyperplane. In comparison to other sampling methods such as simple random sampling and other statistical methods, Latin Hypercube sampling (LHS) aims to cover the maximum range of values present in the design space and the values are restricted to a certain number of dimensions. Therefore, this technique is beneficial to reduce bias and errors during sampling.

2.4 Clustering

Clustering is a grouping method that is used to identify a set of common points or subgroups within a dataset to analyze and understand the underlying patterns within the data. Similar points can be grouped based on a measure of similarity which can be based on euclidean distance or correlation-based distance. In our thesis, clustering methods are used to group and identify different types of crash pulses and their associated restraint settings after optimization. Two types of clustering methods are used in drawing more insights from the lookup table which includes K-Means Clustering and K Nearest Neighbours.

2.4.1 K-Means Clustering

It is an unsupervised recursive algorithm that divides the dataset into ’k’ number of distinct groups, ensuring that there’s no overlapping of data and each point belongs to one group. The algorithm groups data points within each cluster as close as possible by evaluating the sum of squared distances between the data point and the cluster’s centroid to be minimum (Sammut and Webb, 2011). Figure 2.3 details the formation of clusters with the help of the K-Means clustering algorithm below:

The following steps describe the functioning of the K-Means algorithm :

- Initialize the number of clusters ’k’
- Shuffle the dataset and select k number of points at random
- Perform multiple iterations until there are no change to the centroids.

2.4.2 K Nearest Neighbours

The K Nearest Neighbours (KNN) is a supervised machine learning algorithm that is used for classification and regression tasks. It is an instance-based learning algorithm, meaning that the model memorises the instances from the training data and uses it to make predictions. The algorithm assigns the closest category or label to a data point based on it’s euclidean distance to the nearest cluster or group (Sammut

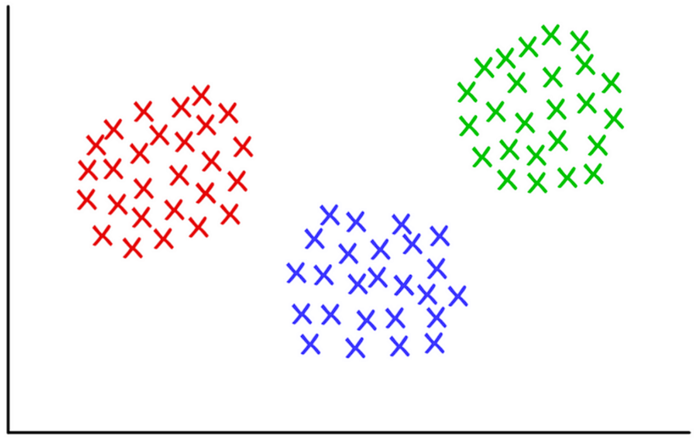


Figure 2.3: Clusters using K-Means (Mayo, 2019)

and Webb, 2011). The steps involved in K-Nearest Neighbours classification are as follows:-

- Select the required number of neighbours 'K'
- Calculate the distance between the new point and training points from a dataset.
- Find the shortest distances and determine the closest neighbours
- Determine the class label or category applicable to the new data point by calculating the averages of the neighbours' values
- Assign the label to the point using the closest point found in the dataset.

K-Means and KNN are simple clustering and classification methods that are easy to implement and categorize relatively smaller datasets. There are higher computational costs and dimensional errors in calculating distances for larger datasets.

The clustering methods have been used in the upcoming chapters in sections 3.4.4 and 4.4.1, detailing the methodology and results obtained from the creation of clusters for different crash pulses and the closest restraint settings for each pulse.

The next section details the methodology used to carry out the work in the thesis.

3
Methodology

With the help of the literature reviewed and the concepts describing the techniques involved in our thesis topic, this section addresses the approach and the working methodology employed in this thesis project.

3.1 Model Simulation and Data

The dataset used in the thesis contained 600 full-frontal simulations involving different impact speeds, overlaps, seating postures, and restraint settings. It was an open dataset provided by Autoliv through which crash pulses were generated using two identical Honda Accord vehicle simulation models. These generated pulses were then applied to a 50th percentile male generic frontal system model, which means that the average male human body model (HBM) based on the mid-point of the total population was utilized during these simulations as shown in Figure 3.1. The standard frontal model was equipped with a driver airbag and a seatbelt with load limiters, capturing data through a carefully formulated design of experiments (DOE). These cases were simulated by Autoliv for head on collisions at crash speeds of 25, 35, 45, 55, 65, and 75 km/h and different vehicle overlap conditions: 50%, 75%, and 100%. The captured data in the DOE was divided into two types of variables – fixed variables and design variables, which described the vehicle crash configuration, seating position, and their corresponding restraint system parameters.

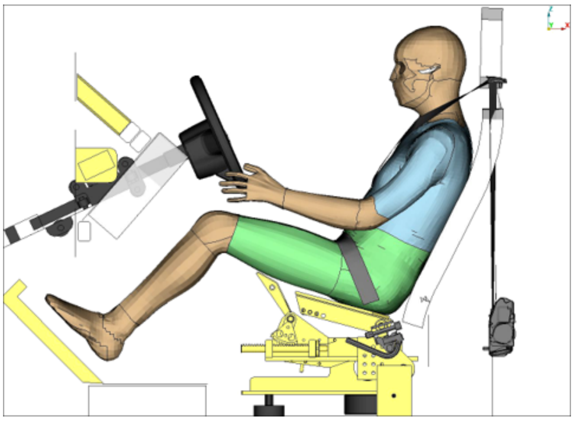


Figure 3.1: SAFER HBM Model restrained with driver airbag and seatbelt Autoliv Sverige AB, 2024

3.1.1 Design Variables

The table shown in 3.1 illustrates the different variables, ranges, description, and their representation in the dataset. The design variables of the DOE include the vehicle crash configuration and seating positions which represent the fixed variables while the decision variables make up the vehicle restraint settings. The design variables will be used in the following sections to predict the kinematic and injury responses for the given dataset.

The fixed variables encapsulate the variables present in the in-crash phase and provides information such as vehicle speed, overlap, and acceleration along with the seating position of the occupants. Alternatively, the decision variables in the dataset include the restraint settings and dictate the response of the vehicle restraint system to protect the occupants during a crash. The fixed and decision variables together represent the input variables in the DOE. Additionally, the input variables consist of discrete values such as vehicle speed, overlap, and seatbelt load limiter settings while the occupant seating configuration and the driver airbag time-to-fire are continuous variables. The ranges for the seating positions and decision variables were provided by Autoliv and describes possible values for the different vehicle velocities and overlaps.

	Variables	Ranges	Description
Fixed Variables	*dv*	*25, 35, 45, 55, 65, 75 km/h*	*Relative crash velocity between target and ego vehicle*
	Overlap	*50, 75, 100 %*	*Percentage of vehicle contact overlap during collision*
	Maximum Acceleration	*Crash dependant*	*Maximum acceleration during the crash event*
	Average Acceleration	*Crash dependant*	*Average acceleration during the crash event*
	Seat Back Angle	*nominal, 10 deg*	*Seat recline angle during event*
	Seat x position	*-50, 50 mm*	*Seat position adjustment relative to pedals*
	Seat z position	*-50, 0 mm*	*Seat height adjustment*
	Slouch (Pelvic Tilt)	*-10, 0 deg*	*The angle of the pelvis with respect to the lower body while seated*
Decision Variables	*Seatbelt LLA LL1*	*53, 63, 73 Nm*	*Stage 1 load limiter; Adjusted by deformation of the torsion bar*
	Seatbelt LLA LL2	*27, 37 Nm*	*Stage 2 load limiter; Adjusted by a deformation of a secondary torsion bar*
	Seatbelt LLA Switch TTF	*30, 50, 70, 90 ms*	*Describes the trigger timing of the load limiter function*
	Driver Airbag TTF	*Ranges between 10 - 30 ms*	*Trigger time of the driver airbag in the event of a crash*

Table 3.1: Design Variables used for Meta models

In the course of this thesis, we made use of fixed variables, that is, the crash scenarios to train the metamodel and further validate the learning of the metamodel as show in figures 4.1, 4.2, and 4.3, respectively. Later on, The Latin hypercube sampling (LHS) algorithm was employed to create unique, non-overlapping combinations for the input variables to represent a set of possible crash configurations and were then updated in the lookup table as described in section 3.4.3 and section 4.4, respectively.

Similarly, the decision variables, that is, the restraint settings were also used along with the fixed variables to train the metamodel to predict the targets and its validation is show in section 4.3.1. During the optimization process these decision variables are obtained by using the fixed samples in LHS by taking the discrete and continuous decision variables described in the gene space of the genetic algorithm in order to generate a set of optimized restraint settings which could minimize the injury sustained by the occupant during a crash.

3.1.2 Kinematic and Injury Responses

The targets represent the kinematic and injury responses that are evaluated after the crash based on the given crash pulse. The main injury criteria in focus for this thesis topic are:

i) Diffuse Axonal Multi-Axial General Evaluation (DAMAGE)

Diffuse Axonal Multi-Axial General Evaluation, abbreviated as DAMAGE, is a type of injury criterion that predicts the level of brain strain in diffusion injuries such as concussion and Diffuse Axonal Injury (DAI). This metric has been recently adopted by the Euro NCAP to help measure traumatic brain injury (TBI). This is because brain injuries can also occur without skull fractures, which the head injury criterion (HIC) did not account for (Gabler et al., 2019). DAMAGE is calculated based on the equations of motion of a second order system with three degrees of freedom, which estimates the extent of brain strain using directional dependant angular acceleration time histories during the moment of impact as depicted by the equation below:

$$\text{DAMAGE} = \beta \max_t \left\{ \left| \vec{\delta}(t) \right| \right\} \tag{3.1}$$

Where β is a scaling factor that describes the maximum displacement of the system model in relation to the maximum brain strain value and $\vec{\delta}$ (t) is the displacement time histories of the brain in all three axes . However, it is more commonly measured as a normalized measure of brain strain between 0 and 1, where 0 implies no damage or strain whereas 1 indicates the maximum strain that the brain can endure before an axonal injury sets in (Gabler et al., 2019).

ii) Minimum Distance From Steering Wheel to Human Body Model

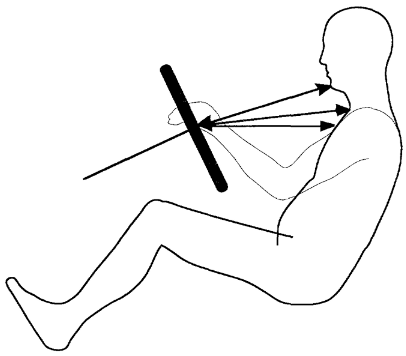


Figure 3.2: Minimum distance between body and steering wheel (Manary et al., 1998)

It is a kinematic response and is useful for assessing the performance of a vehicle restraint system in the event of a crash or collision. In our thesis, we make use of simulations in order to measure the metric. The minimum distance from the body to the steering wheel can be measured in three positions, namely, the minimum distance from the chin to steering wheel center, minimum distance from the manubrium (top of the sternum) to steering wheel center, and minimum horizontal distance from the body to the steering wheel centre as shown in figure 3.2. Overall, while interpreting kinematic responses, the minimum horizontal distance is taken into consideration since it is most critical measure amongst all. A direct contact between the body and steering wheel would result in a major risk of head and chest injuries (Manary et al., 1998). The seat belt load limiter force, trigger time, and airbag trigger time are crucial in this case.

iii) Maximum Resultant T6 Acceleration

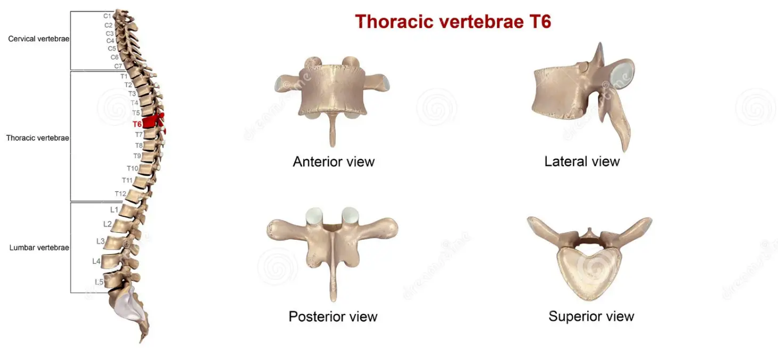


Figure 3.3: T6 Spine Anatomy (Source: Dreamstime, 2024)

The maximum resultant T6 acceleration measures the peak magnitude of acceleration of the 6th vertebra in the thoracic spine which is illustrated in figure 3.3. The T6 acceleration is measured through simulations by assuming the spine as a beam that is subjected to a combination of axial compression and bending moments (Autoliv Sverige AB, 2024). The resultant T6 acceleration determines how well the occupant is restrained during a crash. Higher resultant T6 acceleration equates to a greater risk of injury, so the restraint capacity of the seatbelt load limiter influences the magnitude of the thoracic spine acceleration (Autoliv Sverige AB, 2024).

3.1.3 Correlation Matrix

The correlation matrix facilitates a better understanding of the influence between the design variables and targets in interpreting prediction errors and selection of the appropriate inputs in the metamodel training process. It aids in appropriate feature selection since it helps in visualizing the interdependence between the different variables and the correlation coefficients also provide a general idea of the relationship between the inputs and outputs.The correlation matrix between the design variables and targets is shown in the figure 3.4.

The design and target variables are plotted in the x- and y-axes in the matrix. The extent of correlation between the variables can be determined with the help of the color bar ranging from red to blue. It is noted that the input variables dv, maximum acceleration, and average acceleration exhibit the highest correlation to

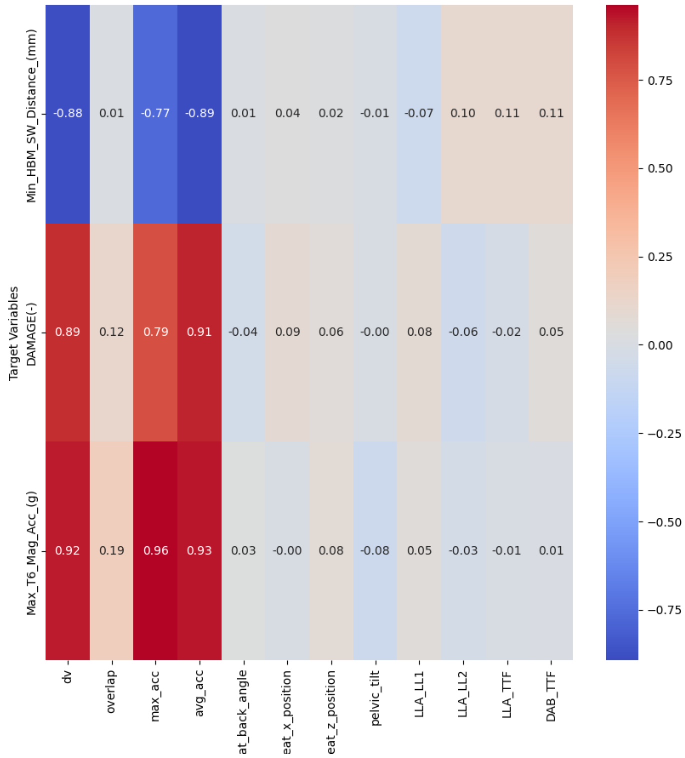


Figure 3.4: Correlation Heatmap of Design and Target Variables

the target variable prediction, while the other variables show little to no influence on the injury risks. The strong negative correlation for the Minimum distance from HBM to steering wheel parameter suggests that the parameter value decreases as the relative vehicle speed and accelerations increase.

It is observed that the fixed variables, particularly describing the vehicle's kinematics, have a strong correlation with the target variables. This is due to the fact that the vehicle speed and acceleration have a large influence on the motion of occupants inside the vehicle during a collision. Therefore, higher velocities and accelerations correspond to higher injury risks for the occupants. Conversely, the remaining input variables depicting the seating position and the restraint settings have a much lower influence on the target variable values. However, the seating position of the occupant and the restraint settings have a significant influence on the activation of the vehicle restraint system, which are crucial in reducing injury risk and safeguarding the occupants during a crash. Hence, these variables are also essential in the metamodel training and optimization in later sections despite the low correlation numbers.

Process Flow (Metamodel)

Based on the process flow model according to Chappell, 2015, the figure 3.5 describes the steps involved in the development of the metamodel for predicting kinematic and injury risk responses as adopted from the process flow. The pre-processing

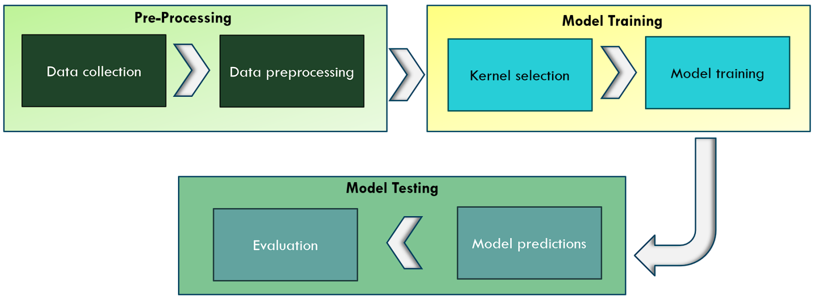


Figure 3.5: Process flow for metamodel development

phase consists of data collection and data preprocessing. The collected data includes the design variables used as the input to the metamodels. Since all the variables have different ranges, they need to be scaled to a common range to allow for easy interpretability by the metamodels.

The model training phase entails the selection of the appropriate kernel function to train the metamodels after splitting and using the scaled data to make predictions. Finally, the model testing phase detail the metamodel predictions and evaluation with the help of error metrics and other methods, which have been described in detail in the results section.

3.2 Data Preprocessing and Normalization

3.2.1 Pre-processing

Preprocessing is an important step in a data-driven process, and the first step involves scouring of the data to look for any outliers, anomalies, unnecessary columns or NaN values present within the dataset. Removal of these unwanted points will ensure minimal interference of external factors in carrying out model training and other related steps. Data preprocessing is further segregated into data scaling and splitting to aid in the metamodel training phase.

3.2.2 Data Splitting

Before training the metamodels, the dataset was split into training and testing sets. The purpose of the splitting is to help identify whether the model understands the data patterns from the training data and is able to generalize similar patterns on unknown data. This is a key aspect of model training in supervised learning applications and is helpful for understanding model performance and error rate. With the help of scikit-learn library, 80% of the DOE dataset was assigned to training data while the remaining 20% was kept as the test set (Scikit-learn library, 2024). Splitting the dataset into the 80:20 ratio ensures sufficient coverage of training data in the metamodels, enabling it to generalize predictions on new data, that is 20% of the actual dataset, with a lesser margin of error compared to other splits such as 60:40 or 70:30 (Gholamy et al., 2018). In addition, setting the test data size to 20%

facilitates improvement of the model's prediction capabilities and reduces chances for overfitting or underfitting the data.

3.2.3 Standardization

It is necessary to scale the data before proceeding with the model training process. Scaling is the process of ensuring that the numerical values of different data points fall within a similar range to ensure uniformity of the data. It enables the metamodels to make better predictions for data within the same scale. Scaling can be performed using different types of methods such as Min_max scaling or standardization to keep the values between a certain range, depending on the data. Scaling methods transform the original values between -1 and 1, although the transformation is dependant on distribution of the dataset and the presence of outliers if any. Standardization scales the data points based on the mean and standard deviation of the data distribution as opposed to using the minimum and maximum values of each column. Transforming the data using the mean and standard deviation ensures that the shape of the distribution is maintained leading to more uniform values. Moreover, standardization is less affected by outliers compared to other methods making it suitable for scaling our dataset (Ahsan et al., 2021). The mean and standard deviation of the input and output are computed with the help of the numpy library before applying the standardization formula to the data. The formula for standardization is depicted by the equation shown below:

$$z = \frac{x - \mu}{\sigma} \tag{3.2}$$

Where x is the original value, while μ and σ represent the mean and standard deviation of x, respectively (Scikit-Learn, 2024c).

3.3 Model Selection and Training

This section describes the selection of the metamodel method and the training process followed.

3.3.1 Metamodel training

Based on the model considerations and studies outlined in sections 1.2.1 and 2.2, the Gaussian process regressor was selected as a suitable model due to a multitude of reasons. It is a non-parametric model, implying that it does not have a fixed functional form and maps the relationship of data points based on the Euclidean distance and covariance between each point. Aside from fitting the curve through a set of points, it also provides a distribution by estimating the mean and variance of each point, accounting for non-linear variations in the dataset (Scikit-Learn, 2024a). The model also works well for small and medium sized datasets and supports both discrete and continuous variables as observed in the design variable table 3.1, motivating our choice accordingly (Goli et al., 2018). The GPR model is governed by covariance functions or kernels, which controls the smoothness of the functions and influences its shape (Hay et al., 2023).

Through the scikit-learn library (Scikit-Learn, 2024a), the three kernels used in the GPR metamodel have been described as shown below:

$$k(\ xi,\ xj) = constant\_value \quad \forall x1, x2 \tag{3.3}$$

Referring to section 2.2.1.1, the constant kernel is used to add or multiply a user-defined constant value with another kernel in order to scale the values according to requirements. Meanwhile, the radial basis function (RBF) kernel or the squared exponential kernel helps determine the value between two points with the help of the length scale parameter (l) and the euclidean distance (d). The equation for the RBF kernel is shown below:

$$k(xi, xj) = exp\left(-\frac{d(xi, xj)^2}{2l^2}\right) \tag{3.4}$$

In addition to these two kernels, the *WhiteNoiseKernel()*, also adds a measurement noise to represent the variability of the data better and help in predictions. It is controlled with the help of the *noise_level* function parameter which determines the amount (variance) of signal noise to be added in the function.

3.3.1.1 Hyperparameter Grid Search

Grid search is a method of identification of hyperparameters or function parameters for each kernel by specifying the range of values in the search space to carry out the search process. The grid search function iterates through a range of possible permutations and combinations of the hyperparameter ranges to select the most applicable values for the kernel function in the GPR metamodel as described below:

$$\text{kernel} = \text{ConstantKernel()} \times \text{RBF()} + \text{WhiteKernel()} \tag{3.5}$$

Hyperparameters	Value Ranges
`n_restarts_optimizer`	1, 5, 10
`alpha`	1e-15, 1e-10, 0.01
`kernel__k1__k2__length_scale`	0.01, 0.1, 10
`kernel__k2__noise_level`	0.01, 0.1, 10

Table 3.2: Hyperparameter grid space

The grid space for the hyperparameters and their ranges are defined in table 3.2 shown above. The n_restarts_optimizer, alpha, length scale, and noise level were the hyperparameters used in RBF, white noise, and the main kernel function, respectively. The hyperparameter selection was undertaken after referring Scikit-Learn, 2024a to determine which parameters positively influenced the model performance.

Moreover, the ranges were selected on a trial-and-error basis by passing the kernel through the GridsearchCV() function to obtain the most suitable values. In addition, the GPR model also contained a function attribute called *kernel_*, which provided the optimized hyperparameters for the kernel function chosen.

Based on the kernel function, the length scale parameter influences the smoothness of the RBF kernel. The noise level parameter indicates the factor of white noise to be added into the metamodel and the values were chosen to select suitable noise levels. Subsequently, alpha and n_restarts_optimizer are the model parameters that influence the overall covariance function. Alpha denotes the variance of gaussian noise in the function and its default value in the function is 1e-10, so the values were chosen in multiples of 10. The n_restarts_optimizer states the number of iterations of the running the built-in optimizer of the GPR to maximize its log-marginal-likelihood (Scikit-Learn, 2024a). Increasing the number of parameters to search exponentially varies the search time in the grid search function. The accuracy of the model using the optimized hyperparameters is assessed with the help of error metrics detailed in section 2.2.1.2 and the results are shown in table 4.1 in section 4.1.2.

3.3.1.2 Learning Curves

Learning curves are plots that illustrate whether the metamodel is learning well or not by comparing the errors obtained using the new data (validation error) against the number of training examples used (training error) against a suitable error metric. Generally, data-driven models produce errors while making predictions. This is attributed to the bias-variance trade-off, where the model either makes an assumption about historical data (bias) while making predictions or displays underlying variations of the model's predictions (variance) once new data is used to train the existing model. The main objective of training the models is to ensure that bias-variance trade-off is minimal without risk of underfitting or overfitting the data (Sammut and Webb, 2011). The model is said to be learning if the generalization error, i.e, the gap between testing and training error, decreases over the number of training examples. Furthermore, the model is said to have a low bias if the training error is less since it is able to fit the data well and vice versa. If the gap between the testing error and training error converges, then the model is said to have low variance and is able to generalize new points. If the variance-bias trade off is high, then the model may require additional training samples.

Learning curves are plotted by making use of a Scikit-Learn, 2024b library *learning_curve()* where the function attributes such as training sizes and cross-validation can be modified according to requirements. It also includes a scoring parameter, which specify the type of error metric to be used. Moreover, standard error metrics such as MAE and MSE are available as negative counterparts, meaning that the error is lesser as the value is closer to zero. The learning curves for all the three metamodels have been illustrated in section 4.1.1 of the results chapter.

3.4 Optimization

Referring to studies in section 1.2.2 and given the diverse types of variables and requirements to minimize multiple objectives in our thesis, the genetic algorithm was chosen due to its flexibility as a global optimization algorithm(Liao et al., 2008). Referring to section 2.3.1, the genetic algorithm works on the principle of natural selection, which means that the population solutions would be evaluated based on their fitness values. The fittest solutions from the population would be selected and operations such as mutation and crossover are performed to create candidate solutions, which would be iteratively evaluated for their fitness until the best solution is found or until a termination criteria is met.

The population solutions in this case refer to the decision variables outlined in section 3.1.1. The solutions for different frontal crash scenarios are obtained after running an optimization routine. They are generated by employing a data sampling algorithm that accounted for the maximum coverage of the parameter space from the dataset. Furthermore, the optimization routine also made use of the trained GPR metamodels to predict the kinematic and injury responses of all the generated samples, which are used to define the fitness function.

Optimization Process Flow

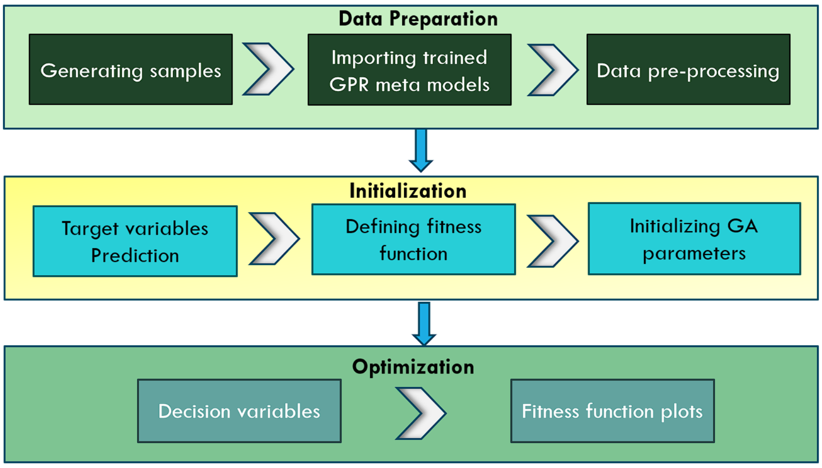


Figure 3.6: Workflow for Optimization

The process flow for optimization is divided into three phases as shown in Figure 3.6. In the data preparation phase, the Latin hypercube sampling (LHS) algorithm is used to generate unique crash configuration samples using the design variables. Later, these samples will be scaled and utilized by the trained metamodels to predict injury risks. Subsequently, the metamodel predictions will be used to define the fitness function of the genetic algorithm with appropriate constraints which are defined later in section 3.4.2. Finally, the optimised decision variables are obtained after initializing the algorithm.

3.4.1 Latin Hypercube Sampling

Referring to section 2.3.2 and the process flow in figure 3.6, the LHS algorithm is used to generate random, unique, non-overlapping values for each variable for in a multidimensional distribution containing 'n' number of dimensions and 'k' number of variables. This method of sampling ensures that the values are well distributed and cover the parameter space with the least number of samples generated. With the help of the scipy library (SciPy, 2024), sixty samples were created for the design variables described, taking the constraints into account, i.e., discrete or continuous variables. These samples are used in running the optimization.

3.4.2 Fitness Function

The genetic algorithm works on the basis of evaluating the fitness criteria to find the fittest solution. The goal of the objective function is to minimize the overall targets, so the fitness function was formulated in such a manner:

Objective: min $f(X)$, where:
- min $f_1(X)$: Minimum HBM-to-steering wheel (SW) distance.
- min $f_2(X)$: Maximum resultant T6 acceleration.
- min $f_3(X)$: DAMAGE

Subject to the following constraints:

$$0\,\text{mm} \leq \text{Min\_HBM\_SW\_Distance} \leq 165\,\text{mm} \tag{3.6}$$

$$0\,\text{g} \leq \text{Max\_T6\_Mag\_Acc} < 150\,\text{g} \tag{3.7}$$

$$0 \leq \text{DAMAGE} < 1 \tag{3.8}$$

Where **X (Decision Variables):**
- Seatbelt LLA LL1
- Seatbelt LLA LL2
- Driver Airbag (DAB) Time-to-Fire (TTF)
- Seatbelt Pretensioner Time-to-Fire (TTF)

The decision variables are constrained by the following bounds:

$$X_{\text{lower}} \leq X \leq X_{\text{upper}}$$

where X_{lower} and X_{upper} are the lower and upper bounds, respectively.

and,

Fixed Variables:
- Change in velocity (Δv)
- Overlap
- Seat back angle
- Seat x-position

- Seat z-position
- Slouch

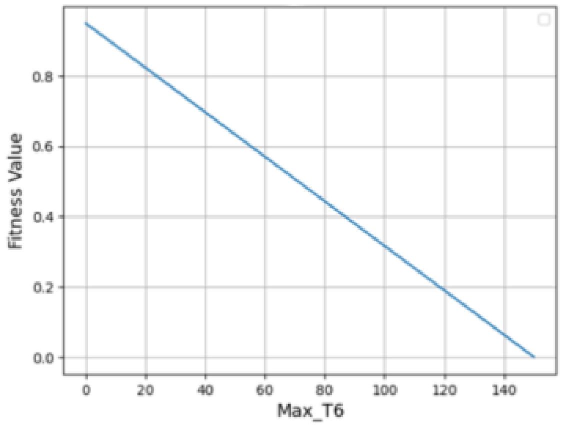


Figure 3.7: Fitness vs Maximum resultant T6 acceleration

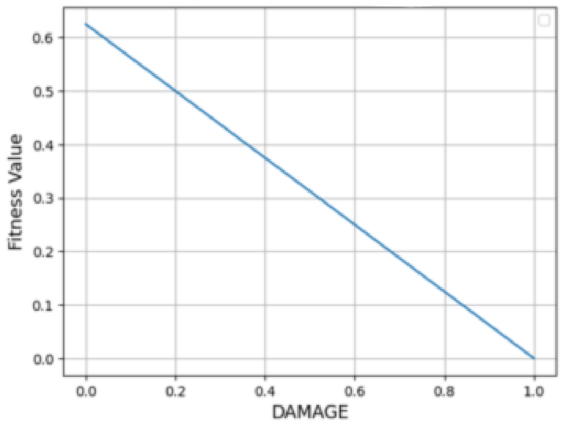


Figure 3.8: Fitness vs DAMAGE

The above equations depicts the fitness function problem in our thesis. The objective is to minimize the overall function f(X) with respect to the three target variables, that is, the minimum distance from HBM to steering wheel f1(X), Resultant T6 acceleration f2(X), and DAMAGE f3(X). The target variables are subject to a set of constraints shown in equations 3.6, 3.7, and 3.8, where a lower value is preferred for each variable. The optimization algorithm in combination with the metamodel use the fixed variables from the DOE and the decision variables from the gene space of GA to predict the three target variables. Later, the target variable predictions are subjected to the constraints defined in the fitness function formulation, which is used by GA to determine the optimal restraint settings based on the highest fitness value of the population sample. According to the constraints defined in the fitness function criteria, a weighted sum method was used to determine the overall fitness value as seen in the following equation:

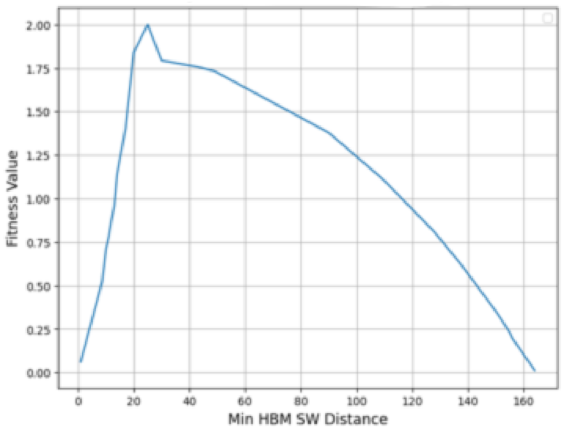


Figure 3.9: Fitness vs Minimum distance to steering wheel

$$\text{Fitness}_{\text{overall}} = w_1 \times \text{Fitness}_{\text{HBM}} + w_2 \times \text{Fitness}_{T6} + w_3 \times \text{Fitness}_{\text{DAM}} \tag{3.9}$$

The weights were assigned according to the importance of the target variables to guide the optimization routine, prioritizing the minimum HBM to steering wheel distance over maximum resultant T6 acceleration and DAMAGE, respectively. This order was established to predict optimal restraint settings for lower injury responses across different frontal collision cases. Based on the maximum resultant T6 acceleration and DAMAGE plots in figures 3.7 and 3.8, the weights were adjusted to maximize individual fitness values for injury risk parameters close to zero, while reducing as the injury risk increases. The equations for calculating the fitness values are demonstrated as:

$$\text{Fitness}_{T6} = 1 - \frac{\text{Pred}_{T6}}{\text{Max}_{T6}} \tag{3.10}$$

$$\text{Fitness}_{\text{DAM}} = 1 - \text{Pred}_{\text{DAM}} \tag{3.11}$$

The fitness values for target variables DAMAGE and Maximum T6 resultant acceleration decrease linearly as per the bounds defined in the constraints. This is because higher values of each variable indicate greater injury risks to the occupant.

In the case of Minimum distance from HBM to steering wheel, setting the maximum fitness value to zero such as DAMAGE and maximum resultant T6 acceleration would result in a strike-through since it would result in a contact between the body and steering wheel, potentially increasing the risk for injury. Observing the plot in figure 3.9, the peak fitness value was set at a safety margin around 21 mm. The safety margin was identified based on the prediction error of the metamodel. The equations for the fitness values of the minimum distance to steering wheel are illustrated below:

$$\text{Fitness}_{\text{HBM}} = \frac{\text{Pred}_{\text{HBM}}}{10} \tag{3.12}$$

$$\text{Fitness}_{\text{HBM}} = \frac{\text{Pred}_{\text{HBM}} - 10}{11.5} \tag{3.13}$$

$$\text{Fitness}_{\text{HBM}} = \frac{\text{Pred}_{\text{HBM}}}{21.5} \tag{3.14}$$

$$\text{Fitness}_{\text{HBM}} = 1 - \left(\frac{\text{Pred}_{\text{HBM}} - 21.5}{\text{Max}_{\text{HBM}} - 21.5}\right)^2 \tag{3.15}$$

Accordingly, the constraints in the weighted sum method maximized the fitness value for minimum body to steering wheel distance from 0 to 21mm in a linear fashion. However, a quadratic equation was used as described in equation 3.11 for distances between 21 mm and 165 mm since values lying close to the safety margin (21 mm) have a high fitness score and steadily decrease towards the maximum bound (165 mm).

3.4.3 Lookup Table

The following section helps in gaining an understanding on the methods used to populate the look up table for adaptive restraint systems.

The sixty randomly generated crash scenarios from LHS were optimized using GA to obtain its respective optimal restraint settings in the optimization phase. Along with this, the trained metamodels were used for predicting the kinematic and injury responses of the occupants. The goal of the lookup table was to map the crash scenarios with its corresponding restraint settings and injury risks in a single structure. The table was developed using a pandas dataframe to store all the fixed inputs, decision variables, and injury risks to enable easy lookup of values. The table format has been broken down into the following categories.

Table 3.3 represents the fixed variables section of the lookup table. The fixed variables are the specific crash scenarios made use of during this thesis. We make use of LHS in order to generate 60 different unique crash scenarios. The sixty randomly generated crash scenarios from LHS were optimized using GA to obtain its respective optimal restraint settings in the optimization phase which are represented by decision variables in table 3.4. Along with this, the trained metamodels were used for predicting the kinematic and injury responses of the occupants which are represented as target variables in table 3.5.

dv	overlap	max_acc	avg_acc	seat_back_angle	seat_x_position	seat_z_position	pelvic_tilt
25	50	10.29	7.39	5.6	2.6	-24.2	-8.7
45	50	25.21	13.61	3.1	34.1	-34	-6.1
75	50	53.81	20.12	9.6	-1.6	-10.6	-3.3

Table 3.3: Fixed Variables Table

LLA_LL1	LLA_LL2	LLA_TTF	DAB_TTF
63	37	50	10.5
53	37	50	27.2
73	27	50	16.3

Table 3.4: Decision Variables Table

Min_HBM_SW_Distance_(mm)	Max_T6_Mag_Acc_(g)	DAMAGE
112.4	14.9	0.1
53.7	30.6	0.4
2.2	83.1	0.6

Table 3.5: Target Variables Table

Cluster
Low Crash Pulse
Medium Crash Pulse
High Crash Pulse

Table 3.6: Cluster Table

On the other hand, the clusters provide information about the different types of crash pulses for each crash scenario. There are three types of pulses, namely, Low, Medium, and High crash pulses and these have been identified based on the velocity of the crash sample, i.e, dv. A clustering algorithm is used to form clusters based on velocity and are illustrated in the lookup table in table 3.6.

The goal of the lookup table is to map the crash scenarios with its corresponding restraint settings and injury risks in a single structure. The table was developed using a pandas dataframe to store all the fixed inputs, decision variables, and injury risks to enable easy lookup of values. The next section details the grouping methods applied to the table to better differentiate between crash scenarios. The result of the lookup table is shown in section 4.4.

3.4.4 Clustering

Clustering methods are applied to contextualize crash samples from the lookup table, helping to understand how restraint settings affect each crash case . Referring section 2.4, two types of grouping methods were used to classify the data from the lookup table, using K-Means and KNN. The crash cases were clustered using K-Means based on crash velocities. According to the correlation matrix, the design variable Δv was strongly correlated with the target variables. So, based on the ranges of

Δv, crash cases were classified into 'Low Crash Pulse', 'Medium Crash Pulse', and 'High Crash Pulse' enabling a clear visualization in the plots shown below:

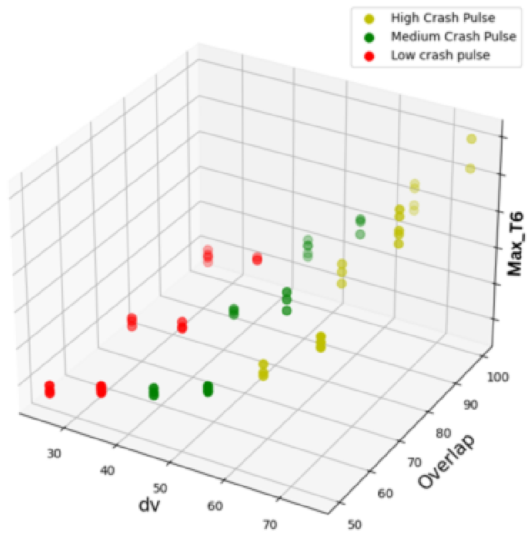


Figure 3.10: Maximum resultant T6 acceleration cluster plot

Figure 3.10 illustrates a 3D plot between Δv, overlap, and Maximum resultant T6 acceleration. The low, medium, and high crash pulses are represented by red, green, and yellow data points on the graph, with the target values increasing proportionally with relative velocity.

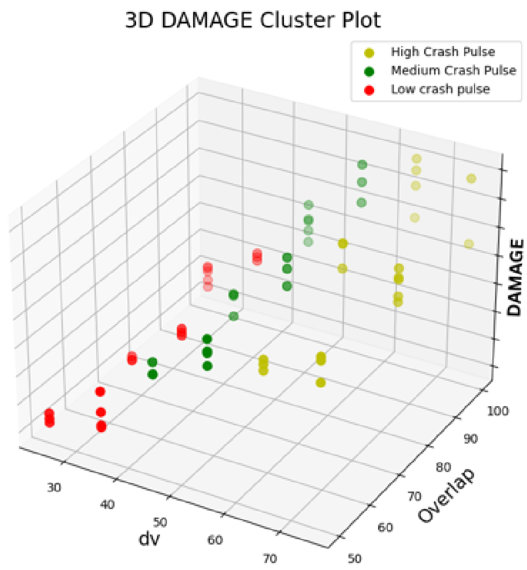


Figure 3.11: Damage cluster plot

In figure 3.11, DAMAGE follows a similar relationship to Δv and overlap, although the data points are more widespread, suggesting an influence of overlap in the plot for the same velocities.

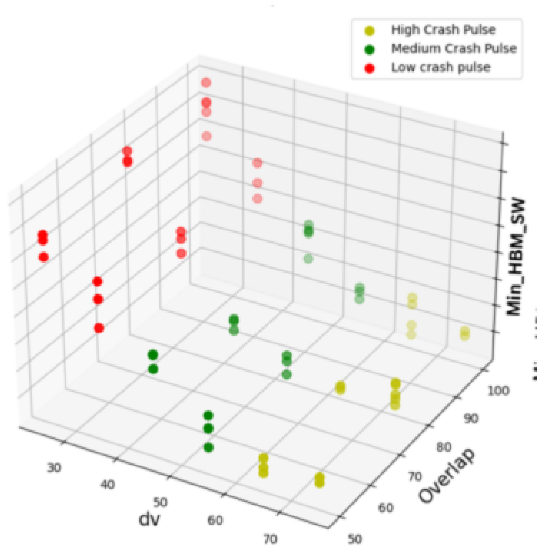


Figure 3.12: Minimum HBM to steering wheel distance cluster plot

In figure 3.12, The Minimum distance from HBM to steering wheel is negatively correlated with Δv and overlap and decreases with higher velocities. Additionally, low minimum head to steering wheel distances can be observed in some higher crash pulses, possibly resulting in a strikethrough.

This approach is also helpful in classifying crash points on the 3D plots and is demonstrated in the results section.

4
Results

This section describes the results.

4.1 Metamodel Validation

The validation has been carried out in two parts. In the first stage, we have compared the predicted values against the actual targets from the DOE on a scatter plot and assessed the model learning with the help of learning curves. The metamodels were validated to understand how well it learns from unknown data (test data) and can emulate the trends from the simulation. Additionally, the accuracy of the metamodels also influences the overall optimization process and the subsequent validation stage. In the second stage, the kinematic and the injury responses predicted by the metamodels after optimization are compared against actual simulations using the optimal restraint settings from GA. Additionally, safety and simulation engineers from Autoliv also suggested potential real-world restraint settings using their own domain knowledge and engineering judgement to account for external factors in real life situations. This is because practical data may not be complete, so it is essential to take a decision that balances engineering principles and limitations to achieve the required objective. Finally, the injury and kinematic responses obtained after running simulations using the real-world settings have been compared against simulations using optimized settings and metamodels. This is done to assess whether the optimization approach can aid in future design and operation of adaptive restraint systems.

4.1.1 Learning Curves:

Based on the process flow in section 3.1, the model evaluation stage is used to characterize the performance and accuracy of the GPR metamodels after training and prediction. The models have been trained on the design variables from the DOE, encompassing the fixed and decision variables such as vehicle speed, acceleration, seating configuration, and the restraint settings. Moreover, the most suitable hyperparamaters for the kernel function are determined using a combination of the GPR model attributes and grid search before the training process. The learning curve function uses some of the training and testing instances to plot their respective errors against a scoring metric for a set number of iterations.

Referring section 3.3.1.3, the learning curves for each of the metamodels are plotted to evaluate the model performance and to compare the varying learning rates to understand the influence of the target variables. The testing and training errors were plotted against the negative mean squared error with five-fold cross validation to check for model convergence over a fixed number of training samples. The negative mean square error was chosen since it helps in identifying outliers in the data, magnifying the error due to the presence of the squared term in the equation. In contrast to a standard MSE measurement, the negative MSE used in learning curves penalizes errors that have larger negative values, for example an error value of -5 is higher compared to -2. This implies that errors closer to 0 have better accuracy. The learning curves for each of the target variables are shown below:

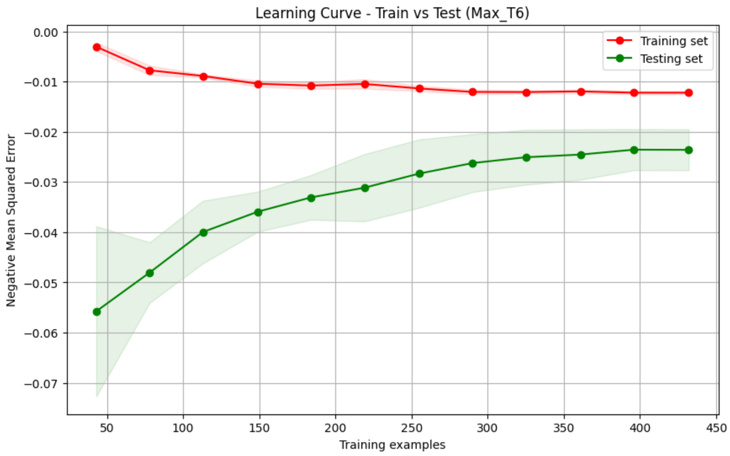


Figure 4.1: Learning curve for metamodel T6

In figure 4.1, the training and testing errors lie between -0.01 and -0.03, with the testing error converging to the training error as the number of samples are increased. Since the value is close to zero, the model performance is quite good. We can also observe higher variance in the testing error, indicating uncertainty of the model to known data.

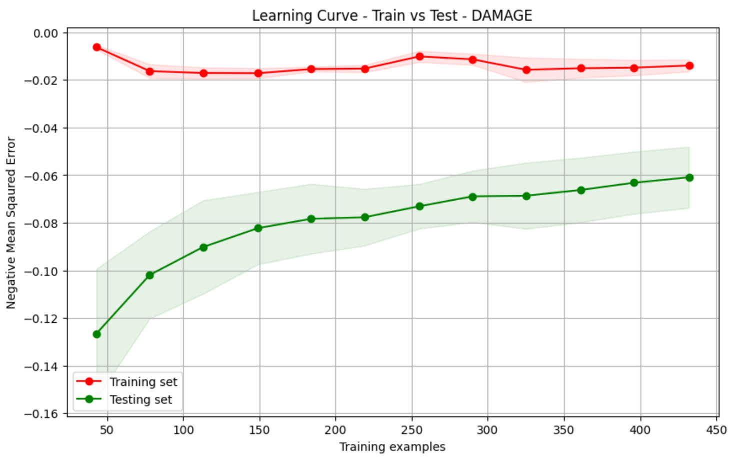


Figure 4.2: Learning curve for metamodel Damage

Figure 4.2 depicts the learning curve for the metamodel predicting DAMAGE. Model learning can be seen as the testing error reduces gradually until the 450th iteration. However, the error margin is higher compared to Maximum resultant T6 acceleration, with the values lying between -0.01 to -0.08, with a varied learning rate and higher variance, implying marginally higher model uncertainty. However, since the values are still close to 0, the model performance is favourable.

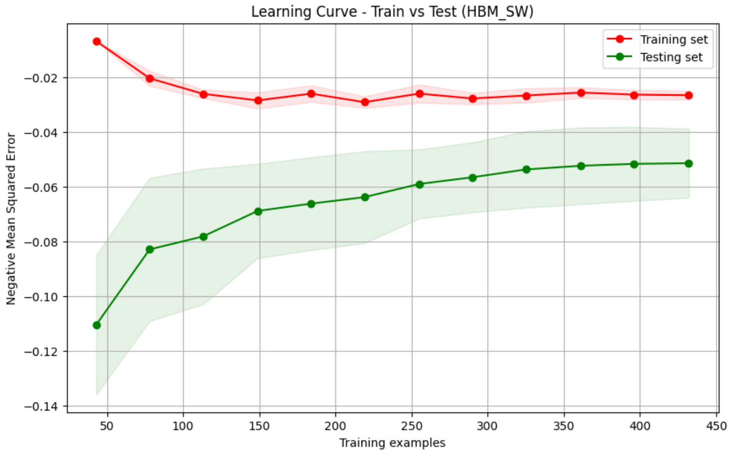


Figure 4.3: Learning Curve for metamodel Min_HBM_SW_Distance

According to figure 4.3, the testing error converges to training error with the model learning well. The error margin is lower compared to DAMAGE, but higher than Maximum resultant T6 acceleration, although a higher variance is observed in the testing error compared to other targets, indicating larger uncertainty in predicting the Minimum HBM to steering wheel distance. Despite this, the model performance is still favorable since the error margin is close to zero (between -0.01 and -0.08).

4.1.2 Actual vs Predicted Plots

In this section, the error between the predicted values from the metamodel and the actual test values are compared to evaluate the predictive capability of the models.

Error Metrics	Max_T6_Acc	DAMAGE	Min_HBM_SW
MAE	0.10	0.17	0.21
MSE	0.02	0.05	0.06
RMSE	0.14	0.21	0.25
R2	0.97	0.94	0.93

Table 4.1: Metamodel regression metrics for three targets

The plots illustrates the distribution of the predicted values against the diagonal reference line representing the test values. In figure 4.4, the predictions closely follow the reference line with a minimal error margin for almost all values, implying good accuracy for the metamodel predicting 'Maximum resultant T6 acceleration'. The predicted values follow a similar trend in figure 4.5 in the case of 'DAMAGE' with

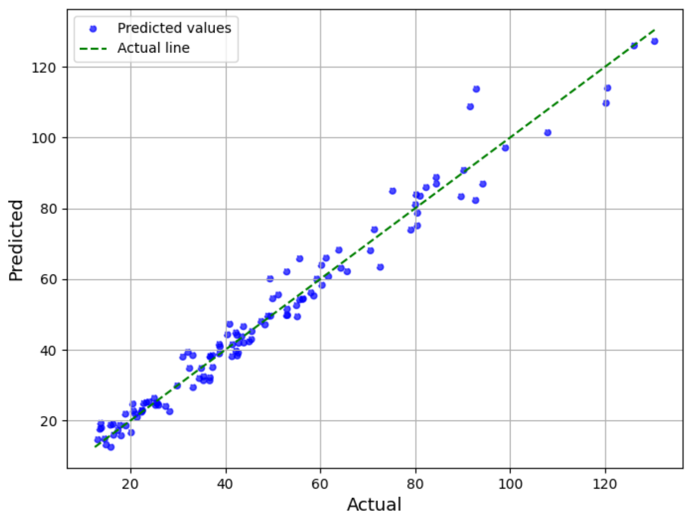


Figure 4.4: Actual vs Predicted Test Values - T6

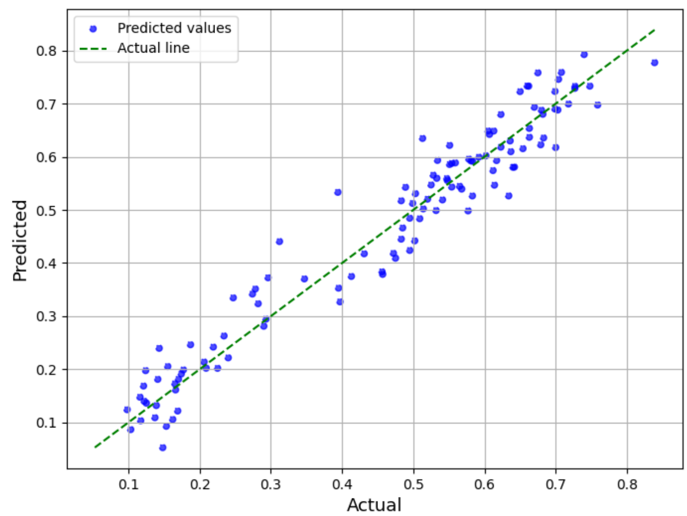


Figure 4.5: Actual vs Predicted Test Values - Damage

the values following the reference line, albeit with a marginally higher error margin. In figure 4.6, an accumulation of data points towards the origin were observed in the case of 'Minimum distance from HBM to steering wheel'. This can be attributed to the occurrence of a strikethrough event, implying possible physical contact between the head and steering wheel in extreme cases. Hence, the overall error in predicting the 'Minimum HBM to steering wheel distance' is higher compared to the other targets, although the predictions are still close to the actual test values. Table 4.1 outlines the error differences between the three target variables.

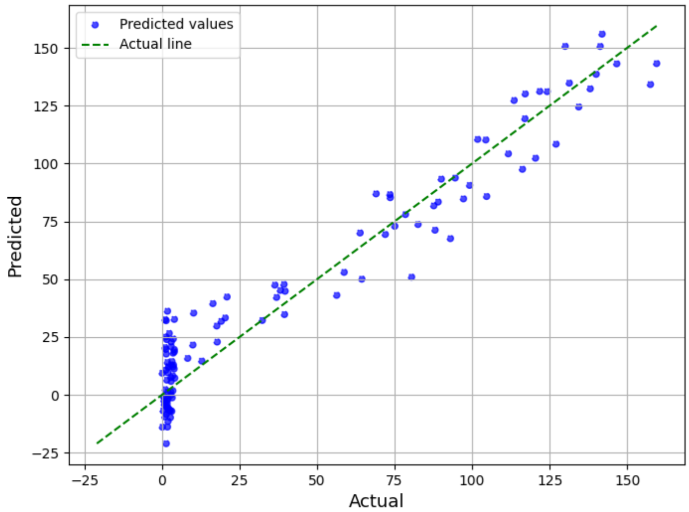


Figure 4.6: Actual vs Predicted Test Values - Min_HBM_SW_Distance

4.2 Optimization

4.2.1 Generation vs Fitness

Referring to section 3.4, the optimization of the restraint settings with the help of GA was performed on a set of sixty fixed samples sourced from the Latin hypercube sampling algorithm and one such sample with its optimized results has been showcased as an example for understanding in figures 4.7, 4.8 , and 4.9 , respectively.

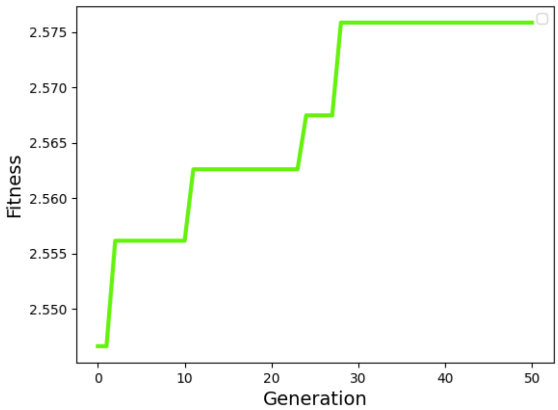


Figure 4.7: Fitness value vs Generations

The genetic algorithm evaluated the crash sample on the basis of a fitness function (Refer 3.4.2), prioritizing a higher fitness value for lower target value predictions according to the constraints defined in the function. In the fitness vs generation plot shown in figure 4.7, the genetic algorithm converges to the optimal solution of the sample in the 30th generation with a maximum fitness value of 2.575 out

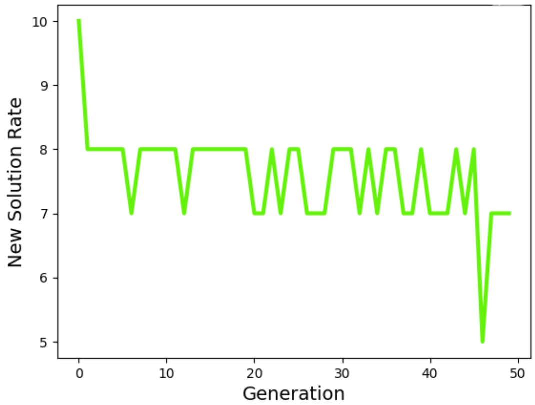


Figure 4.8: New Solution Rate for Fixed Sample

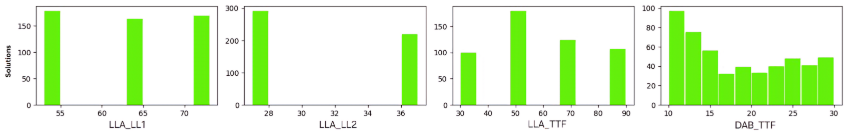


Figure 4.9: Gene Distribution

of 3. Figure 4.8 illustrates the number of new solutions found by the GA across the number of generations or iterations. The gene distribution plot illustrates the number of solutions generated per population by the optimization for each gene. As shown in Figure 4.9, the genes represent the decision variables derived from the design space, which are the solutions produced following the optimization process.It is interesting to note that the gene distribution is marginally skewed, implying that the genetic algorithm is more biased towards a certain range for some genes, especially DAB_TTF towards the lower bounds. The distribution is dependent on the population sample chosen and is indicative of its heuristic nature in finding the optimal solution.

The graphs help in understanding how the genetic algorithm determines the optimal solution by iterating through multiple combinations of the genes to find the solution with the highest fitness value.

4.3 Optimization Validation

The six samples shown in table 4.2 were sourced from LHS algorithm and the design space, with three samples chosen from each set, respectively. These samples contain different crash velocities and the injury and kinematic responses after optimization were compared with the Finite Element (FE) simulations conducted on the virtual human body model (HBM).

dv	overlap	max_acc	avg_acc	seat_back_angle	seat_x_position	seat_z_position	pelvic_tilt
25	100	14.82	8.34	5	0	-45	-9
35	50	25	10.56	1.1	8.2	-45.2	-8
45	100	32.74	14.5	9	40	-25	-1
55	50	29.51	16.49	0.7	29.3	-25	-2.9
65	100	55.25	20.97	1.8	-32	-5.6	0.6
75	75	70.84	22.74	3	20	-5	-5

Table 4.2: Optimization samples

4.3.1 Simulation Analysis

For the six samples shown in table 4.2, the optimal restraint settings and targets were obtained after carrying out the optimization and predictions from the metamodel, respectively. However, effective validation of the optimized settings is achieved by comparing the injury and kinematic responses (target variables) from the genetic algorithm and the simulations. In order to validate this, the optimized settings for the six samples were applied by Autoliv to the human body model equipped with seatbelts, load limiters, and the driver-side airbag. Additionally, technical experts and simulation engineers working with restraint system development used engineering judgement to develop a different set of restraint settings based on currently available restraint systems (state-of-the art) (Weedon, 2017).

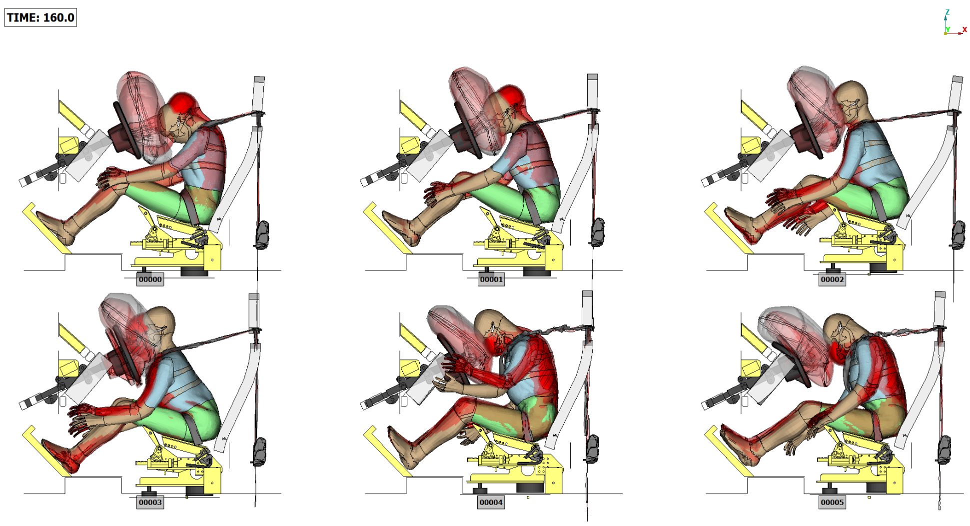


Figure 4.10: Simulations for fixed samples

Autoliv also conducted FE simulations on the human body model using the state-of-the art (SOTA) restraint settings and the target values were compared with metamodels and simulations with the optimized settings. Table 4.3 details the comparison between the three methods. The optimization and other simulations use the same

Settings	Decision variables
Optimization	Yes
Simulations using optimization values	Same as optimization
State of the Art	Engineering judgement

Table 4.3: Simulation table

set of six fixed samples as described in Table 4.2. The decision variables are generated after optimizing the fixed samples in GA. Along with the decision variables, we also get the kinematic and injury response predictions from the trained metamodels. Later, the injury responses from the metamodels are compared with two different simulations. The simulation with the optimized settings uses the decision variables from the optimization to obtain the injury responses. Meanwhile, the simulations using the state-of-the-art settings use a different set of decision variables to obtain the injury outcomes. These settings have been provided by Autoliv's simulation and test engineers based on engineering judgement that reflect current vehicle restraint systems as described earlier. Overall, the comparison of injury responses between the three different methods is to validate the optimization approach and find out whether the optimized settings can lower the injury risks for the occupants seated in the vehicle.

Figure 4.10 is a still image from the simulations with the timestamp at 160ms. The crash samples from left to right in the top row represent the first three fixed samples from the table 4.2 , whereas the bottom row portrays the last three samples. Due to differences in crash velocities, distinct variations can be observed in the interactions between the model and the steering wheel. Moreover, the red overlay in each of the crash samples helps to differentiate between SOTA restraint settings from optimized restraint settings. The injury and kinematic responses between metamodels and simulations have been compared in the subsequent subsections.

4.3.1.1 Max_T6 Acceleration

Based on the plot in figure 4.11, the maximum resultant T6 acceleration increases steadily with velocities in the metamodels and simulations. The T6 values using the optimized settings are consistently lower than the SOTA restraint settings in all cases, indicating lower injury risk. The predicted values from the metamodel capture the trends of the simulations and are marginally lesser in all cases except for a dv of 55 km/h, demonstrating good overall accuracy.

.

4.3.1.2 DAMAGE

According to figure 4.12, DAMAGE follows a similar trend to maximum resultant T6 acceleration, with values increasing proportionally with dv. Simulations using optimized settings showcase lower injury risks in low and medium crash pulses compared to SOTA, i.e., for dv between 25 - 55 km/h, but are slightly larger in high

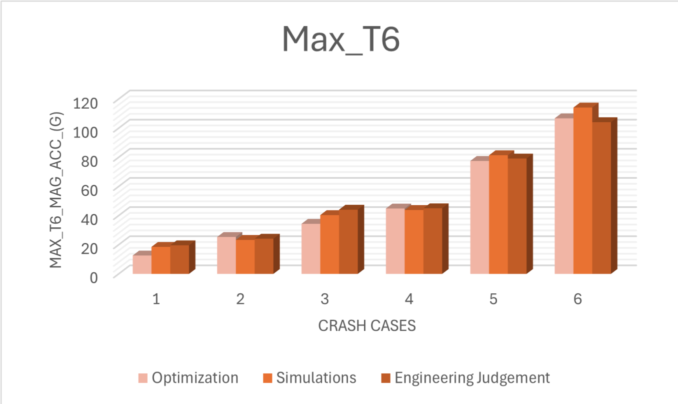


Figure 4.11: Maximum Resultant T6 Acceleration for different crash pulses

crash pulses. The metamodel predictions are also relatively close to the simulations, with slightly higher values for dv of 25, 45, and 65 km/h, respectively.

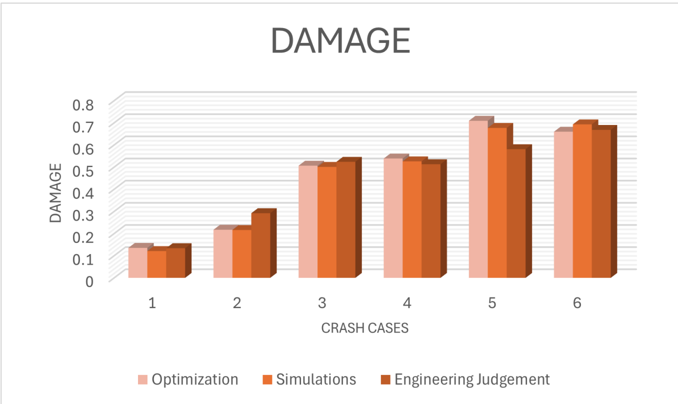


Figure 4.12: DAMAGE for different crash pulses

4.3.1.3 Minimum distance from HBM to steering wheel

The minimum distance from HBM to the steering wheel decreases as dv increases, but the SOTA system shows significantly higher values at low velocities (dv of 25, 35 km/h) compared to metamodels and simulations. Predicted values from the metamodel are lower than simulations and SOTA for dv between 25 and 55 km/h, but are higher in the latter two cases. This discrepancy may be due to prediction errors in the metamodel, particularly at higher velocities. Nevertheless, the metamodel accurately captures the trends observed in simulated and SOTA results, as depicted in figure 4.13.

By comparing the target values of the metamodel with the simulations, the model predictions are quite similar to the responses generated by the simulations for maximum resultant T6 acceleration and Damage, although there are larger differences

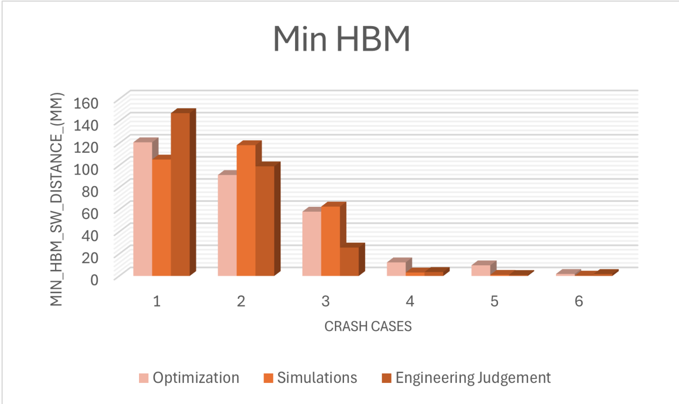
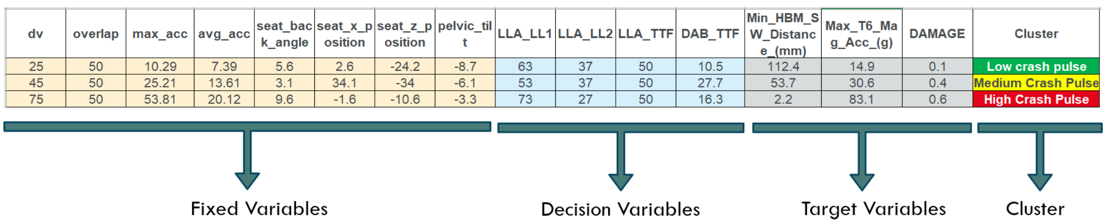


Figure 4.13: Minimum HBM to steering wheel distance for different crash pulses

while predicting the minimum distance from HBM to steering wheel. The predictive capabilities can be further improved by tuning the metamodel and choosing appropriate hyperparameters to reduce errors.

4.4 Lookup Table

As described in section 3.4.4, the data structure for the lookup table has been depicted in the figure 4.14. The table has been divided into four sections with each of them representing the fixed variables, decision variables, target variables, and their associated clusters respectively. Referring section 3.4.5, the crash samples obtained from LHS were classified into different crash pulse types with the help of K-means clustering and added to another column in the lookup table. The table stores the values of the crash samples along with their optimized restraint settings, responses, and cluster types. For an adaptive restraint system, the lookup table can be utilized to find a direct setting depending on the vehicle speed and seating configuration or the closest setting can be obtained from the lookup table to maximize the occupant protection during a crash.



dv	overlap	max_acc	avg_acc	seat_back_angle	seat_x_position	seat_z_position	pelvic_tilt	LLA_LL1	LLA_LL2	LLA_TTF	DAB_TTF	Min_HBM_SW_Distance_(mm)	Max_T6_Mag_Acc_(g)	DAMAGE	Cluster
25	50	10.29	7.39	5.6	2.6	-24.2	-8.7	63	37	50	10.5	112.4	14.9	0.1	Low crash pulse
45	50	25.21	13.61	3.1	34.1	-34	-6.1	53	37	50	27.7	53.7	30.6	0.4	Medium Crash Pulse
75	50	53.81	20.12	9.6	-1.6	-10.6	-3.3	73	27	50	16.3	2.2	83.1	0.6	High Crash Pulse

Figure 4.14: Visual Representation of the Lookup Table

4.4.1 Clustering

According to section 3.4.4, the LHS samples were classified into 'Low Crash Pulse', 'Medium Crash Pulse', and 'High Crash Pulse' depending on the relative velocities using K-Means. Velocities of 25km/h and 35 km/h belong to Low Crash Pulse', 45km/h and 55 km/h are assigned the 'Medium Crash Pulse', whereas 65km/h and 75km/h belong to 'High Crash Pulse'.

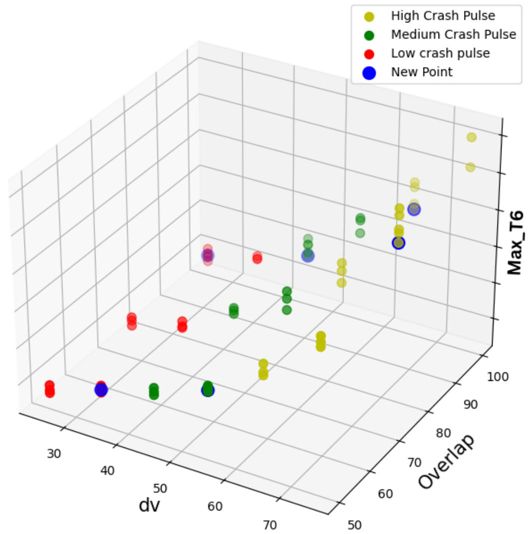


Figure 4.15: Max_T6 Cluster Plot

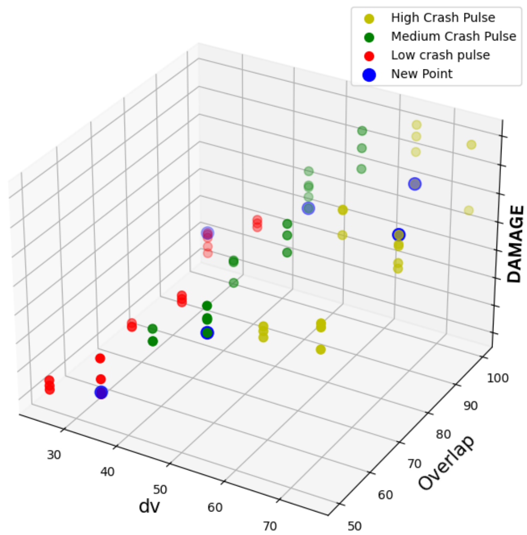


Figure 4.16: Damage Cluster Plot

However this approach can be also be extended to group new data points. For example, if a set of crash scenarios are not present in the lookup table, the opti-

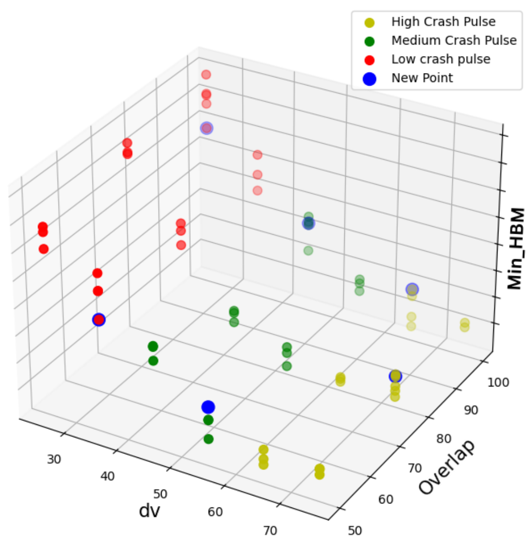


Figure 4.17: Minimum Human body model to Steering Wheel distance Cluster Plot

mization process helps in finding out the most optimal restraint settings while the grouping method identifies the clusters and crash types for each point. Later, these points are added to the lookup table to enhance the speed and performance of the restraint system when encountering a similar crash scenario in the future. This was demonstrated by adding the six crash samples shown in Table 4.2 to the existing lookup table containing the LHS samples. To easily differentiate between the existing samples, the new data points can be visualized with the blue points with a dark circular outline in the 3D plots for each of the targets shown accordingly:

From the plots in figures 4.15, 4.16, and 4.17, the new points have been scattered according to the position of different clusters in each of the targets. The approximate location of these points are useful in determining the severity of the pulse and aids in selection of appropriate restraint settings.

The KNN classification method was utilised in the lookup table to obtain the nearest restraint settings for any given data point according to its design variable configuration. This is favourable if an exact setting cannot be directly retrieved from the lookup table.

5
Discussion

This thesis is focused on occupant protection systems such as seatbelts and airbags, which restrain the occupant and aim to mitigate the risk of injuries and dissipate the forces away from the body during a collision.

The topic is primarily focused on developing a framework to select the most optimized settings in an adaptive restraint system by using machine learning methods and grouping them based on crash pulses. The initial step involved using GPR metamodels to predict the injury and kinematic responses from the dataset. One metamodel for each target prediction was selected and the their individual performances were evaluated with the help of learning curves and calculating the error between the actual values and predicted values. Based on the learning curves in figures 4.1, 4.2, and 4.3, the testing and training errors were the lowest for T6 acceleration followed by Damage and the distance from HBM to steering wheel. The model convergence or learning rate also followed a similar order, however, all the errors were very close to zero while measuring the negative mean squared error, indicating good accuracy. The metamodel predictions were also compared with actual test values indicated by the diagonal reference line in figure. The models predicting T6 acceleration and DAMAGE demonstrated better accuracy when compared to minimum distance from HBM to steering wheel, although they have good overall accuracy. This is illustrated in the error metrics table 4.1, where MAE, MSE, and RMSE are close to zero for all models. Meanwhile, the R2 scores are above 0.9 for all metamodels. The accumulation of points towards the origin indicate the possibility of strikethrough for some crash configurations.The metamodels were evaluated by comparing the actual vs predicted values and assessing the model learning with the help of learning curves, we can infer that the metamodel can capture the trends of the simulations with a good accuracy as validated by the error metrics table. Hence, the trained metamodels were effective in predicting the kinematic and injury responses for a set of frontal crash scenarios, doing so with lesser computational requirements and time compared to simulating each crash scenario. Other studies such as ref and ref have also used metamodels in relation to the development of adaptive restraint systems. However, it is also important to consider prediction errors and model complexities while interpreting results from metamodels in practical applications.

These metamodels were used in combination with an optimization algorithm to select the most optimal restraint settings for a crash scenario. The presence of uncertainties and non-linear relationships with inputs and outputs motivated the need for selecting a stochastic based optimization method. Initial optimization approaches were carried out using Bayesian Optimization and Simulated Annealing, however, both methods did not account for handling discrete and continuous variable types in the restraint settings and minimizing multiple objectives using the metamodels. This motivated the selection of the genetic algorithm (GA) to carry the optimization. A space-filling sampling algorithm, latin hypercube sampling, was utilised to generate sixty unique crash samples for the optimization. The results presented one sample the plots that were obtained after running the optimization. Figure 4.7 illustrates the number of generations taken by the algorithm to maximize the fitness value of the sample, which was 2.575 in the 30th generation. The subsequent figure 4.8 depicts the number of possible solutions found out by GA during each generation. The fluctuations in the new solutions graph highlight that the algorithm is heuristic in nature, meaning that the approach to finding the solution is random and is dependent on the optimization parameters like mutation rate, crossover, and number of parents chosen, amongst others. This also plays an influence in selecting the appropriate values of the genes highlighted in the gene distribution plot in figure 4.9. For this sample, a bias is observed towards lower values in the case of driver airbag time-to-fire as possible solutions, demonstrating the nature of the optimization algorithm. Referring to the fitness function in section 3.4.2, the fitness value is a weighted sum of the target variables that have been normalized.

5.0.1 R.Q.1 Can model-based optimization of restraint system settings help in reducing the injury outcomes of occupants during a crash?

In this section, we analyse how the optimized restraint settings can help in reducing the injury outcomes of occupants in different crash scenarios. The restraint system settings include the parameters that control the seatbelt load-limiter torque, timing, and the driver airbag timing in a vehicle safety system. In a conventional safety system, the load-limiter and driver airbag are semi-adaptive, implying that the settings are adjustable to certain extent for some crashes, but a default load-limiter torque, timing, and airbag time-to-fire setting is used for other collision scenarios. However, adaptive restraint systems can adjust the restraint settings based on the external environment, aiding in better occupant protection. Therefore, the restraint settings need to be optimized to reduce injury risk to occupants for different collision scenarios.

In the optimization phase, six crash scenarios which included three samples generated from the LHS algorithm and the remaining from the DOE, representing various crash velocities were optimized using a combination of GA and metamodels to find the optimal restraint settings. The results from optimization were validated by evaluating the target values predicted by the trained metamodels against the injury responses generated after running simulations on the HBM by Autoliv. One set of simulations was carried out by using the optimized restraint settings, while technical experts and simulation engineers working in restraint system development at Autoliv developed a different set of restraint settings for currently available restraint systems (SOTA) for the same six samples. The metamodels, simulations using optimized settings, and simulations with SOTA settings have been visualized and compared in the bar plots.

Each of the target variables have been plotted against dv, due to a strong correlation between them as seen on the correlation matrix. Referring figure 4.11 , the Maximum Resultant T6 acceleration values are quite similar between the metamodels and the simulations for all crash velocities. The SOTA values (shown in red) are marginally higher than the other two methods in all cases, with a more noticeable spike seen at 45 km/h. The simulations using the optimized settings are consistently lower than SOTA, suggesting lower T6 accelerations for all velocities. The predictions from the metamodel follow the responses by the simulations closely with slightly lower values except at 55 km/h, demonstrating good accuracy by the metamodel.

In contrast to T6 acceleration, distinct variations can be observed in the case of DAMAGE in figure 4.12. It follows a similar relationship to T6 acceleration where the target values increase proportionally along with the crash velocities. However, noticeable spikes can be observed in the metamodel prediction at 25, 45, and 65 km/h compared to the simulations, suggesting model errors during prediction. Meanwhile, the simulations using SOTA settings are higher than other two methods between 35 to 55 km/h, with a sharp increase at 45 km/h, possibly due to an incorrect setting. Using the optimized settings, the risk for damage is lower in the low and medium crash pulses between 25 and 55 km/h.

Based on the figure 4.13, the minimum HBM to steering wheel distance decreases with velocity due to the movement of the occupant in relation to the crash velocity. When comparing the three methods, the simulations with the SOTA settings have significantly higher values compared to the metamodels and the simulations using the optimized settings. This could be attributed to stiffer settings, restraining the movement of the occupant at low crash velocities (25 and 35 km/h). In contrast, there's a sharp decrease in HBM to steering wheel distance using SOTA settings from 45 km/h onwards, potentially indicating a strikethrough at higher velocities. The metamodel and simulations with the optimized settings closely follow each other, although the predictions are still higher at 65 and 75 km/h despite the simulations clearly indicating a contact between the head and steering wheel. This can be explained by the metamodel error while predicting minimum HBM to steering wheel distance in extreme cases.

The bar plots compared the injury and kinematic responses from the trained metamodels after optimization, simulations using optimized settings, and simulations using the state-of-the-art settings. From the analysis, we observed that DAMAGE and Maximum resultant T6 acceleration were lesser for velocities between 25-55 km/h in the simulations with optimized settings as compared to state-of-the-art settings. Moreover, the distance between the HBM and the steering wheel were larger for simulations with the optimized settings compared to SOTA. This indicates potentially lower injury risks using the optimized settings compared to real-world equivalent settings for low and medium crash velocities as validated by the simulation results. However, the injury risks were greater at higher velocities for optimized settings, implying that the model may require additional tuning of parameters and data to resolve the error differences.

5.0.2 R.Q.2 How can the complexity of restraint systems be reduced by grouping and visualizing optimized restraint settings for different crash scenarios?

Through metamodel development and optimization techniques, the optimal restraint settings have been obtained for the sixty LHS samples. They have also been validated by analysing the target responses between the metamodel and simulations. The optimized settings produced lower injury risks demonstrated by lower T6 acceleration, Damage, and Minimum HBM to steering wheel distance values for low and medium crash velocities. This was further enhanced by developing a lookup table using a dataframe to map the fixed crash samples to its corresponding restraint settings and injury responses. The lookup table reduces computational time by removing the need for using the metamodel and optimization to predict injury responses and restraint settings from scratch.

Two types of grouping methods – K-Means clustering and K Nearest Neighbours classification were utilised on the lookup table to categorize the crash samples. With the help of K-Means, three clusters based on crash pulse (velocity) were used to group the samples into low, medium, and high crash pulses for easy understanding. Alternatively, the KNN classification approach was developed to retrieve the nearest possible restraint settings for a given crash sample by calculating the distances between the design variables of the crash sample and the lookup table. This approach complements the value lookup from the table to obtain a direct restraint setting for a crash sample if present in the table. The clusters can be visualized using the 3D plots illustrated above in figures 4.15, 4.16, and 4.17 for the three target variables. The low, medium, and high pulse clusters can be identified by the red, green, and yellow points. Additionally, new points can also be added to the existing table and discerned from the other points. As a demonstration, the six samples used in optimization validation are highlighted by blue points representing different crash pulses. Hence, this approach helped in simplifying the complexity of restraint systems.

However, this is a fixed lookup table and is applicable to a single vehicle model in frontal collisions. It needs to be adapted to work with other vehicle models and constraints. Moreover, the computational time and resources required for grouping methods increases significantly depending on the size and dimensions of the dataset being used.

5.0.3 R.Q.3 How can lookup tables be combined with other machine learning based tools in selecting the optimized restraint settings for crash scenarios?

Figure 5.1 describes a potential implementation of the cluster-based lookup table for a frontal collision scenario. The scenario involves two vehicles travelling in opposite directions on the same road at different speeds. Based on the situation in the ego vehicle, we can determine the vehicle parameters from the electronic control unit

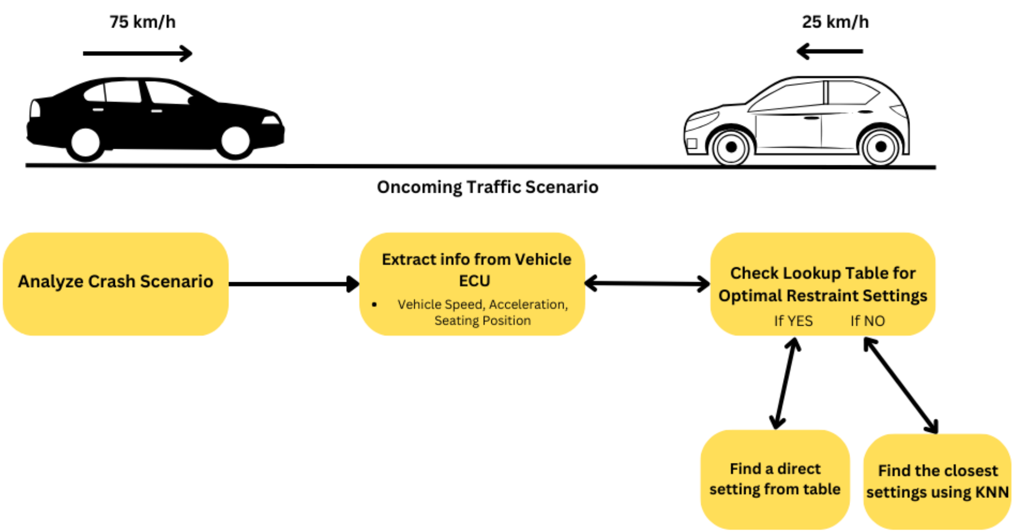


Figure 5.1: Lookup table operational use case

(ECU) of the vehicle such as relative velocity, acceleration, and seating position of the occupant. These parameters can be used to search for the optimal restraint settings in the lookup table using two ways – i) Find an optimal setting directly from the table or ii) Utilize the KNN classification approach to obtain the nearest possible restraint setting in the table for the concerned collision scenario. Once the most optimal settings have been determined, the information is relayed back to the ECU to actuate the seatbelt and airbag of the vehicle with the specific parameters. Through this application, the cluster-based lookup table can be adopted in a practical scenario to quickly retrieve the optimal restraint settings and protect the occupants in the event of a collision. However, it is to be noted that this approach has been developed for a single vehicle model and appropriate modifications need to be carried out to suit other model characteristics and crash types.

The thesis work demonstrated the applicability of machine learning tools in the development of adaptive restraint systems. The combination of metamodels using GPR and the genetic algorithm aided in the estimation of kinematic and injury outcomes and obtaining optimal restraint settings for various frontal collision scenarios at different velocities and vehicle overlaps. The injury predictions from the metamodel after optimization were validated by running simulations for six distinct crash configurations using the optimized and real-world equivalent restraint settings, respectively. The injury outcomes with the optimized settings were smaller in low and medium velocities, but greater at higher velocties. This can be attributed to the error difference present in the metamodel and optimization and may require further parameter tuning and adjustments to improve its accuracy. The metamodel and optimization approach was helpful in capturing the patterns from simulations with lower comptutational resources and time as also demonstrated by other studies. The optimized restraint settings along with the corresponding crash scenarios and injury responses were stored in a lookup table to avoid recalculating values and enable quick retrieval of settings at any given instance. Subsequently, a K-means classification method was utilized to distinguish the crash configurations and restraint settings based on Low, Medium, and High crash pulses. This was done to simplify the table complexity and to easily identify existing and new crash configurations based on the previous criteria. The function of the lookup table is to quickly search and select the most suitable restraint settings for a particular crash situation through a direct lookup. However, a KNN classifier was used to find out the nearest restraint settings for a given crash configuration and type if a direct solution was not available in the lookup table. A potential use-case of the cluster-based lookup table was discussed with a hypothetical head-on collision scenario between two vehicles. The lookup table helps to speed up the search and application of the optimal restraint settings to protect the occupants involved in the crash. The approach also showcases a potential as a tool for design and development of an adaptive restraint system.

The research was limited to a single vehicle model and the results vary between each vehicle model due to the differences in vehicle structures, body, and mass, etc. This would require model re-training and parameter adjustments to account for the vehicle characteristics. Moreover, the influence of human variability such as body type, gender, and age plays a significant influence on the interaction between the occupant and restraint system. Designing adaptive restraint systems also needs to account for other collision types such as rear-collisions, far-side impacts, pedestrian impacts and other road users since crash angles and other factors can influence the restraint settings. As safety norms are expected to become more stringent in the long run with the evolution of test standards by Euro NCAP and other regulatory bodies, leveraging newer technologies to build safer and more capable restraint systems to protect occupants is essential in the future.

6
Conclusion

The GPR based metamodels were trained on a dataset of 600 unique crash scenarios with defined design variables to predict the corresponding kinematic responses and injury values. This provided us with a mathematical representation of the relationships that exists between them.

By optimizing the decision variables with the help of the trained metamodels the best set of restraint settings for specific crash scenarios are generated. The validation of the target variables predicted by the restraint settings with that of simulated results, gave us a deeper understanding of the predictive capabilities of the GPR and GA combination and also helped us identify regions that require improvement. These values were then stored in the lookup table. The lookup tables provided us with an efficient and quick method to retrieve the most optimal restraint settings for a specified crash scenario.

Furthermore, a grouping method was established which helped us classify and visualize the optimal restraint settings based on the delta velocity, overlap and injury values. This groups similar scenarios together, thus reducing the complexity and improving the visualization of the restraint settings. This can potentially be useful when designing a new restraint system for occupant safety in the automotive safety field as it allows for quick retrieval of data without additional computational time required. The lookup table can be utilized in an operational use case in real-time crash scenario in retrieval of the most optimal restraint settings by getting the vehicle parameters, external factors like vehicle speed and acceleration. In addition, it can also take into account the occupant seating position to offer adequate protection and minimize the injury risks experienced by the occupant.

7

Future Scope

This thesis project has demonstrated the potential for optimizing restraint settings and making use of lookup tables to reduce the injury outcomes of an occupant during a crash. However, there are several opportunities for enhancement hence future work should aim at :

- **The refinement of the GPR and GA could produce better results**. A much more in depth exploration of these concepts and making use of the combination of various parameters present in the GPR like hyperparameters and the different modules present in GA could help in generating more accurate predictions of restraint settings.
- **More detailed classification and clusters** could be implemented to the lookup table to further improve restraint settings for complex crash scenarios. The current lookup table has been clustered into different types of crash pulses presenting a simplified approach. For more complex restraint systems, multiple clusters based on specific restraint settings or additional crash types could be implemented into the lookup table to suggest optimal settings for more specific scenarios.
- **Adaptive lookup tables which make use of real-time data** in order to dynamically adjust to the crash scenarios could present another potential advancement from this thesis. The static lookup table in our thesis contain a set of crash scenarios which are helpful in some cases but may not fully account for the variability of incidents or collisions in the real world. By continuously updating the lookup table based on crash scenarios, the vehicle restraint system can utilize the most recent information to potentially enhance its capabilities.
- **Expanding the study to cover a broad range of restraint settings** could help improve the robustness of the proposed optimization technique in this thesis. For example, optimizing seat belt pretensioner restraint settings in order to assign the appropriate amount of slack to the seat belt of the occupant based on their physique to restrict their movement and to ensure appropriate positioning for the airbag deployment could be part of the future work of this thesis. Furthermore, this lookup table approach could also be expanded to other active safety systems in order to help early prevention of collision.

Bibliography

Ahsan, M. M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, *9*(3). https://doi.org/10.3390/technologies9030052

Athias, V., Mazzega, P., & Jeandel, C. (2000). Selecting a global optimization method to estimate the oceanic particle cycling rate constants.

Autoliv Sverige AB. (2024). Autoliv sweden website [[Online; accessed 2024-07-09]]. https://www.autoliv.com

Bengtsson, L. E. (2012). Lookup table optimization for sensor linearization in small embedded systems.

Chappell, D. (2015). *Introduction to azure machine learning*. Chappell & Associates.

Deng, Y., Lin, G., & Yang, X. (2020). Multifidelity data fusion via gradient-enhanced gaussian process regression. *arXiv preprint arXiv:2008.01066*.

Dreamstime. (2024). Thoracic vertebrae illustration [Stock illustration of thoracic vertebrae, sourced from Dreamstime]. https://www.dreamstime.com/stock-illustration-thoracic-vertebrae-t-group-twelve-small-bones-form-vertebral-spine-upper-trunk-image78412616

Eppinger, R., Gabler, H., Laituri, T., & O'Connor, R. (2017). *Evaluation of an adaptive restraint system* (tech. rep. No. DOT HS 812 432). National Highway Traffic Safety Administration. Washington, DC. https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/812432_adaptiverestraintreport.pdf

Gabler, L. F., Crandall, J. R., & Panzer, M. B. (2019). Development of a second-order system for rapid estimation of maximum brain strain. *Annals of biomedical engineering*, *47*, 1971–1981.

Gad, A. F. (2023). Pygad: An intuitive genetic algorithm python library. *Multimedia Tools and Applications*, 1–14.

Genie, D. S. (2024). Latin hypercube sampling vs monte carlo sampling [Accessed: 2024-09-19].

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. *Int. J. Intell. Technol. Appl. Stat*, *11*(2), 105–111.

Goli, S. A., Far, B. H., & Fapojuwo, A. O. (2018). Vehicle trajectory prediction with gaussian process regression in connected vehicle environment⋆. *2018 IEEE Intelligent Vehicles Symposium (IV)*, 550–555. https://doi.org/10.1109/IVS.2018.8500614

Guardiola, C., Pla, B., Blanco-Rodriguez, D., & Cabrera, P. (2013). A learning algorithm concept for updating look-up tables for automotive applications. *Mathematical and Computer Modelling*, *57*(7-8), 1979–1989.

Hay, J., Schories, L., Bayerschen, E., Wimmer, P., Zehbe, O., Kirschbichler, S., & Fehr, J. (2023). Application of data-driven surrogate models for active human model response prediction and restraint system optimization. *Frontiers in Applied Mathematics and Statistics*, *9*, 1156785.

Hayes-Roth, F. (1984). The knowledge-based expert system: A tutorial. *Computer*, *17*(09), 11–28.

Hu, Z., Du, D., & Du, Y. (2020). Gaussian process-based spatiotemporal modeling of electrical wave propagation in human atrium. *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2602–2605. https://doi.org/10.1109/EMBC44109.2020.9176468

Joodaki, H., Gepner, B., & Kerrigan, J. (2021). Leveraging machine learning for predicting human body model response in restraint design simulations. *Computer Methods in Biomechanics and Biomedical Engineering*, *24*(6), 597–611.

Joodaki, H., Gepner, B., Lee, S.-H., Katagiri, M., Kim, T., & Kerrigan, J. (2021). Is optimized restraint system for an occupant with obesity different than that for a normal bmi occupant? *Traffic injury prevention*, *22*(8), 623–628.

Jung, E. S. (1988). Development of an expert systems for ergonomic workplace design and evaluation. *Proceedings of the Human Factors Society Annual Meeting*, *32*(11), 617–621. https://doi.org/10.1518/107118188786762423

Lemmen, P., Fagerlind, H., Unselt, T., Rodarius, C., Infantes, E., & van der Zweep, C. (2012). Assessment of integrated vehicle safety systems for improved vehicle safety. *Procedia-Social and Behavioral Sciences*, *48*, 1632–1641.

Liao, X., Li, Q., Yang, X., Zhang, W., & Li, W. (2008). Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and multidisciplinary optimization*, *35*, 561–569.

Lochrie, G., Doljevic, M., Nona, M., & Yoon, Y. (2021). Anti-windup recursive least squares method for adaptive lookup tables with application to automotive powertrain control systems [Modeling, Estimation and Control Conference MECC 2021]. *IFAC-PapersOnLine*, *54*(20), 840–845. https://doi.org/https://doi.org/10.1016/j.ifacol.2021.11.276

Manary, M. A., Flannagan, C. A., Reed, M. P., & Schneider, L. W. (1998). Predicting proximity of driver head and thorax to the steering wheel. *Proceedings of the 16th International Technical Conference on Experimental Safety Vehicles, Paper*, 98–S1.

Mayo, M. (2019). *An introduction to clustering algorithms* [[Online; accessed 2024-07-09]]. https://www.kdnuggets.com/2019/04/introduction-clustering-algorithms.html

Nguyen-Tuong, D., Seeger, M., & Peters, J. (2008). Computed torque control with nonparametric regression models. *2008 American Control Conference*, 212–217. https://doi.org/10.1109/ACC.2008.4586493

Plevris, V., Solorzano, G., Bakas, N. P., & Ben Seghier, M. E. A. (2022). Investigation of performance metrics in regression analysis and machine learning-based prediction models. *8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022)*.

Roser, M. (2023). Causes of death globally: What do people die from? *Our World in Data*.

Sammut, C., & Webb, G. I. (2011). *Encyclopedia of machine learning*. Springer Science & Business Media.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), 160.

Scikit-Learn. (2024a). *Gaussian process regression* [[Online; accessed 2024-07-09]]. https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html

Scikit-Learn. (2024b). Scikit-learn learning curves [Accessed: 2024-09-19]. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html

Scikit-Learn. (2024c). *Standardization using scikit-learn* [[Online; accessed 2024-07-09]]. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Scikit-learn library. (2024). *Train test splitting* [[Online Resource; accessed 2024-07-09]]. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html#train-test-split

SciPy. (2024). *Latin hypercube sampling* [[Online; accessed 2024-07-09]]. https://scipy.github.io/devdocs/reference/generated/scipy.stats.qmc.LatinHypercube.html#scipy.stats.qmc.LatinHypercube

Shields, M. D., & Zhang, J. (2016). The generalization of latin hypercube sampling. *Reliability Engineering System Safety*, *148*, 96–108. https://doi.org/https://doi.org/10.1016/j.ress.2015.12.002

van den Hove, M., Mlekusch, B., & Müllerschön, H. (2005). Fe-simulation based optimization of an adaptive restraint system considering multiple front-crash load cases using ls-opt. 5th ls-dyna forum, bamberg, 2005.

Van Ratingen, M., Williams, A., Anders, L., Seeck, A., Castaing, P., Kolke, R., Adriaenssens, G., & Miller, A. (2016). The european new car assessment programme: A historical review. *Chinese journal of traumatology*, *19*(02), 63–69.

Wang, J. (2023). An intuitive tutorial to gaussian processes regression. *Computing in Science & Engineering*.

Wang, W.-C., Cheng, C.-T., Chau, K.-W., & Xu, D.-M. (2012). Calibration of xinanjiang model parameters using hybrid genetic algorithm based fuzzy optimal model. *Journal of Hydroinformatics*, *14*(3), 784–799.

Weedon, J. S. (2017). Putting engineering judgment in conversation with engineering communication. *2017 IEEE International Professional Communication Conference (ProComm)*, 1–8.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning* (Vol. 2). MIT press Cambridge, MA.

Yeh, I., Kachnowski, B., & Subbian, T. (2005). An expert system for vehicle restraint system design. *SAE transactions*, 1543–1553.

DEPARTMENT OF INDUSTRIAL AND MATERIAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY